



Universidad  
de Alcalá

## Grado en Ingeniería en Sistemas de Información

### Arquitectura y Diseño de Sistemas Web y C/S

### Práctica 2: Visualización de cabeceras del protocolo HTTP

**Docente:**

D. Roberto Barchino Plata

D. Javier Albert Segui

**Alumnos:**

Razvan Virgil Craciun

Daniel González González

Carlos Mayor Heras

Javier Arenas Manzanares

Rubén Merino Sacristán

Octubre 2021

## ÍNDICE

1.	Introducción.....	3
2.	Visualización de cabeceras del protocolo HTTP. ....	3
3.	Ejercicio – Análisis de cabeceras. ....	4
3.1	Cabeceras Universidad de Alcalá de Henares. ....	4
3.2	Cabeceras Roll20. ....	9
3.3	Cabeceras El País. ....	12
3.4	Cabeceras El Corte Inglés. ....	16
3.5	Cabeceras nivel20. ....	20
3.6	Cabeceras lolesports. ....	23
4.	Consideraciones finales. ....	27

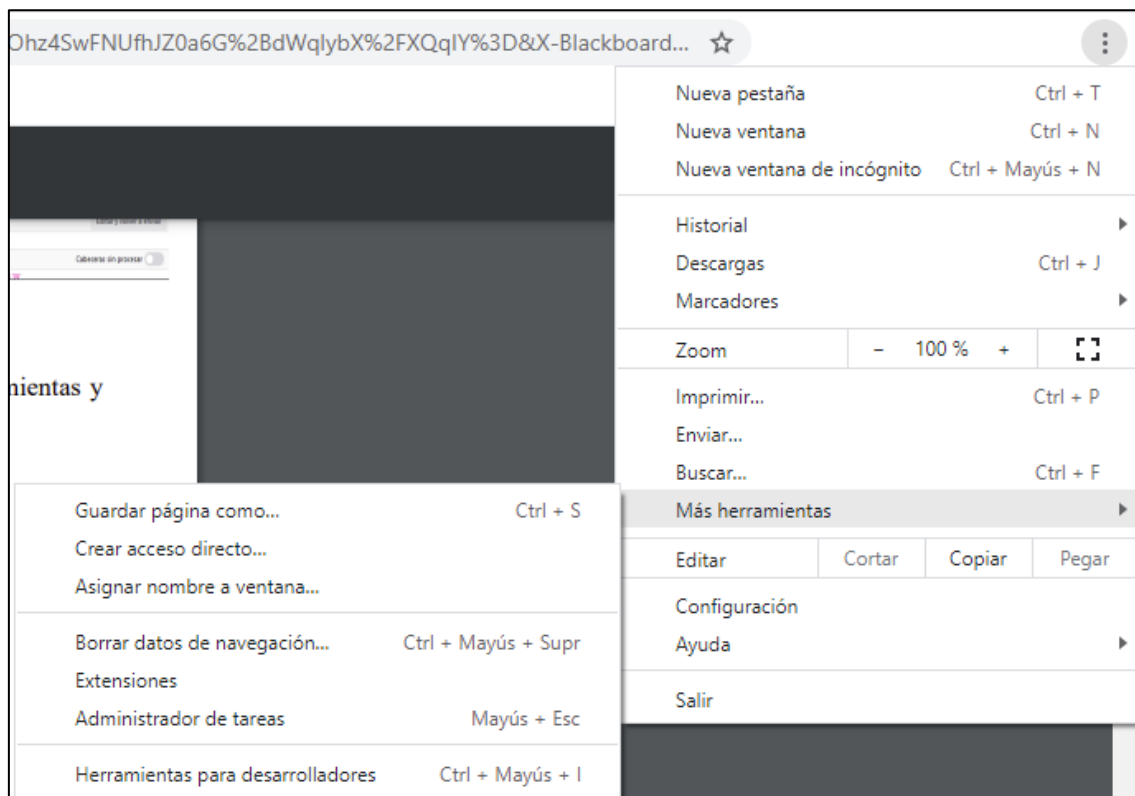
## 1. Introducción.

En el siguiente documento se va a proceder a la utilización de las herramientas para desarrolladores de los navegadores internet, como pueden ser Google Chrome o Firefox, para posteriormente visualizar las cabeceras del protocolo HTTP en cada transacción. Una vez hecho esto, se escogerán y analizarán en detalle 3 cabeceras de tipo GET y 3 cabeceras de tipo POST de varias páginas web.

## 2. Visualización de cabeceras del protocolo HTTP.

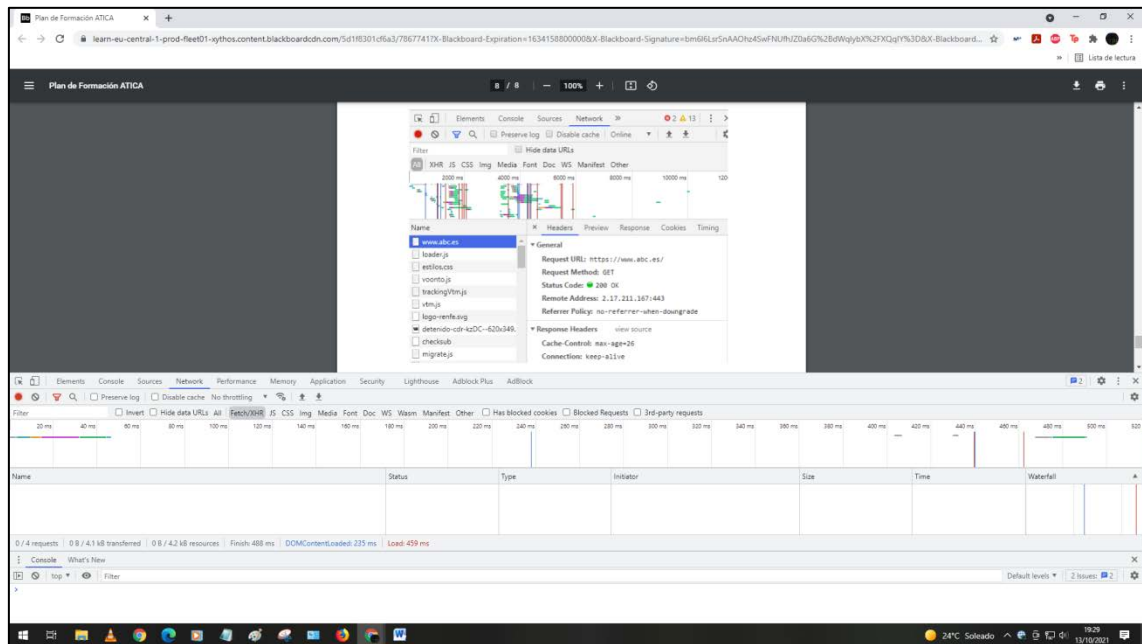
Para poder visualizar las cabeceras de una página web cualquiera, tenemos que recurrir a las herramientas que hemos mencionado en el punto anterior. En nuestro caso se ha utilizado Google Chrome como navegador principal.

Lo primero que debemos hacer es ir a la esquina superior derecha de la pestaña principal de Chrome y abrir las opciones de personalización. A continuación hacemos click en “Más herramientas” y, por último, en “Herramientas para desarrolladores”.



También podemos acceder con el atajo de teclado “Ctrl + Mayús + I”, como se indica en la imagen anterior.

Una vez hecho esto, nos aparecerá una consola de información de las transacciones HTTP de la página web en la que nos encontramos.



Para poder ver las cabeceras, deberemos ir a la pestaña llamada “Network” y, dentro de ella, “Fetch/XHR”.

### 3. Ejercicio – Análisis de cabeceras.

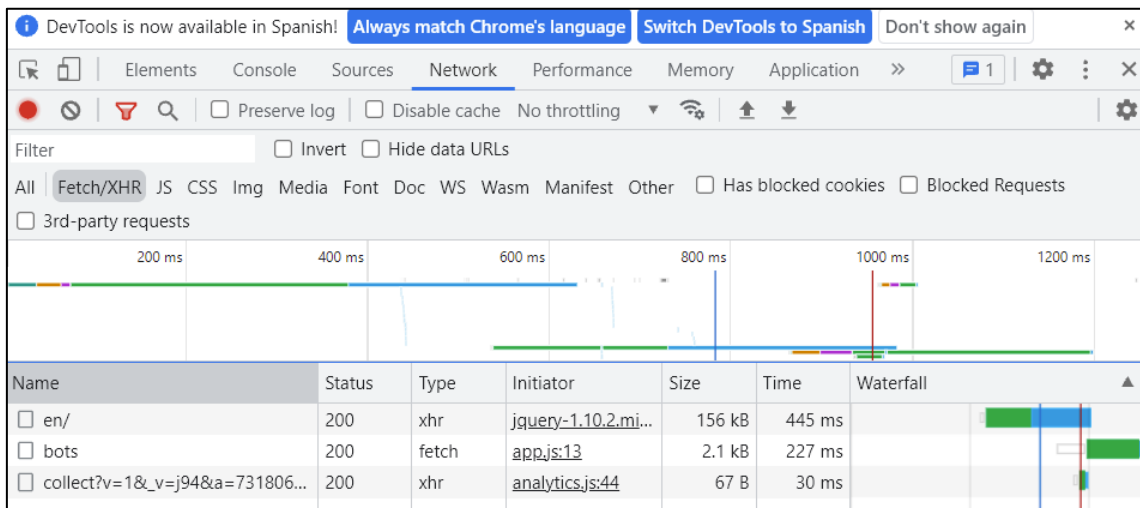
Dentro de los parámetros de cada una de las cabeceras veremos tres grandes grupos, los cuales analizaremos en detalle: el grupo *General*, que nos da una visión general, los *Encabezados de respuesta*, que son aquellos con los que contesta la página, y los *Encabezados de solicitud* que son con los que iniciamos dicha llamada HTTP.

#### 3.1 Cabeceras Universidad de Alcalá de Henares.

Primero accederemos a la página solicitada: <https://www.uah.es/es/>



Una vez dentro, procederemos a visualizar las cabeceras que tiene la propia página web desde la opción de “*Herramientas para desarrolladores*” proporcionada por Google. Una vez dentro, accedemos al apartado *Network* o *Red* dependiendo del idioma que tengamos definido y accedemos a *Fetch/XHR*, donde podremos observar las siguientes cabeceras:



En este punto podemos ver tres cabeceras, cada una perteneciente a una tarea en específico. Nosotros nos centraremos en hablar de la segunda llamada en concreto.

## 1. Llamada “bots”:

▼ General
<b>Request URL:</b> https://api.1millionbot.com/api/public/bots
<b>Request Method:</b> GET
<b>Status Code:</b> 🟢 200 OK
<b>Remote Address:</b> 35.233.2.188:443
<b>Referrer Policy:</b> strict-origin-when-cross-origin

▼ Response Headers	<a href="#">View source</a>
<b>Access-Control-Allow-Origin:</b> *	
<b>Cache-Control:</b> no-store, no-cache, must-revalidate, proxy-revalidate	
<b>Connection:</b> keep-alive	
<b>Content-Encoding:</b> gzip	
<b>Content-Type:</b> application/json; charset=utf-8	
<b>Date:</b> Tue, 12 Oct 2021 09:28:11 GMT	
<b>Expires:</b> 0	
<b>Pragma:</b> no-cache	
<b>Retry-After:</b> 1	
<b>Server:</b> nginx/1.18.0 (Ubuntu)	
<b>Strict-Transport-Security:</b> max-age=15552000; includeSubDomains	
<b>Surrogate-Control:</b> no-store	
<b>Transfer-Encoding:</b> chunked	
<b>Vary:</b> Accept-Encoding	
<b>X-Content-Type-Options:</b> nosniff	
<b>X-DNS-Prefetch-Control:</b> off	
<b>X-Download-Options:</b> noopen	
<b>X-Frame-Options:</b> SAMEORIGIN	
<b>X-RateLimit-Remaining:</b> 13	
<b>X-RateLimit-Reset:</b> Tue Oct 12 2021 09:28:12 GMT+0000 (Coordinated Universal Time)	
<b>X-XSS-Protection:</b> 1; mode=block	

▼ Request Headers	<a href="#">View source</a>
<b>Accept:</b> */*	
<b>Accept-Encoding:</b> gzip, deflate, br	
<b>Accept-Language:</b> es-ES,es;q=0.9	
<b>Authorization:</b> API-KEY 60553d58c41f5dfa095b34dd	
<b>Connection:</b> keep-alive	
<b>Content-Type:</b> application/json	
<b>Host:</b> api.1millionbot.com	
<b>Origin:</b> https://www.uah.es	
<b>Referer:</b> https://www.uah.es/	
<b>sec-ch-ua:</b> "Chromium";v="94", "Google Chrome";v="94", ";Not A Brand";v="99"	
<b>sec-ch-ua-mobile:</b> ?0	
<b>sec-ch-ua-platform:</b> "Windows"	
<b>Sec-Fetch-Dest:</b> empty	
<b>Sec-Fetch-Mode:</b> cors	
<b>Sec-Fetch-Site:</b> cross-site	
<b>User-Agent:</b> Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/94.0.4606.81 Safari/537.36	

Una vez accedemos a nuestra cabecera nos aparecerán de tres tipos, estas son:

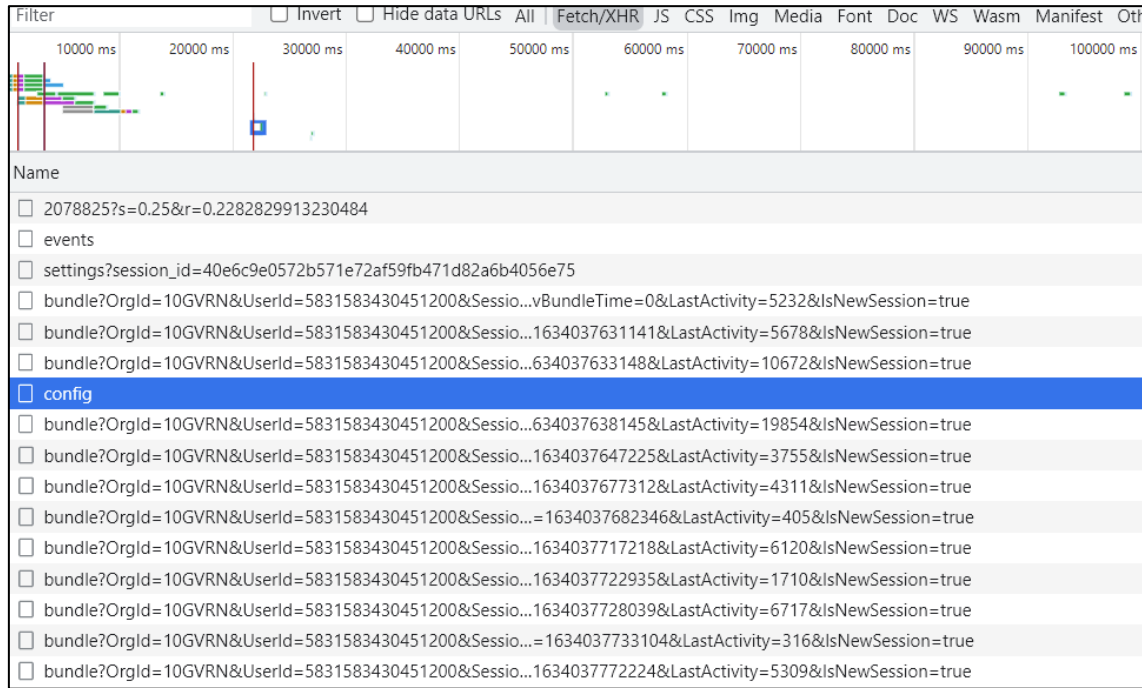
- **General:** En ella podemos sacar información de vital importancia, en nuestro caso podemos ver que nuestra cabecera es de tipo GET (*Request method*), que la URL de la cabecera es <https://api.1millionbot.com/api/public/bots> (*Request URL*), que su estado de respuesta HTTP es satisfactorio, puesto que se agrupa entre el rango de 200 a 299, acompañado de un *OK* y un círculo verde (*Status Code*), su dirección remota es 35.233.2.188:443 y los datos de referente de HTTP determinados “*strict-origin-when-cross-origin*” nos indica que no hay ningún referente.
- **Response Headers (Encabezados de Respuesta):**
  - Para iniciar una petición para los servidores con respuesta, usamos la primera cabecera *Origin: Access-Control-Allow-Origin*.
  - **Cache-Control:** Esta cabecera nos permite conocer las políticas de caché, en nuestro caso:
    - **No-Store:** La respuesta puede que no se almacene en cualquier caché.
    - **No-Cache:** La respuesta puede estar almacenada por cualquier caché, incluso cuando esta petición no es cacheable.
    - **Must-Revalidate:** Cada vez que un recurso pasa a estar obsoleto, el cache no ha de usar esta copia obsoleta sin ser validada anteriormente por el servidor en el que se ha originado.
    - **Proxy-Revalidate:** Parecido al anterior, pero para caches que son compartidos (proxies). Son ignorados por caches privados.
  - **Connection:** Controla el tipo de conexión, en nuestro caso esta conexión se mantiene activa (*Keep-Alive*).
  - **Content-Encoding:** Usada para comprimir el media-type y permite decodificarlo al cliente. En nuestro caso tenemos *gzip*, originalmente es el formato de UNIX y es usado por *Lempel-Ziv coding* con un CRC de 32 bits.
  - **Content-Type:** Nos dice el tipo de contenido que aparece en la petición de POST o PUT que será retornado al cliente. *application/json; charset=utf-8*
  - **Date:** Nos proporciona la fecha y la hora a la que hemos realizado la petición. En nuestro caso: Tue, 12 Oct 2021 09:28:11 GMT
  - **Expires:** Contiene la fecha y la hora en la que se considera la respuesta caducada. El valor 0, como el que tenemos en nuestro caso, representa una fecha en el pasado, lo cual significa que ya ha expirado el recurso.
  - **Pragma:** Usado para la compatibilidad con versiones anteriores de las memorias caché HTTP/1.0 en las que HTTP/1.1 no está presente. En nuestro caso, no-cache significa que han hecho una solicitud al servidor de origen para su validación antes de que se libere una copia en caché.
  - **Server:** Nos proporciona información acerca del software usado por el servidor original, el cual se encarga de nuestra solicitud: *nginx/1.18.0 (Ubuntu)*.
  - **Strict-Transport-Security:** Es una característica de seguridad que permite al sitio web indicar a los navegadores que solo pueden comunicarse con HTTPS. Podemos observar además:

- **max-age=15552000:** Es el tiempo en segundos que el navegador ha de recordar que el sitio solo puede ser accesible usando HTTPS.
  - **includeSubDomains:** Parámetro opcional que nos indica que estas reglas son aplicadas para los subdominios.
  - **Transfer-Encoding:** Especifica la forma de codificación usada para ser segura al transferirse al usuario. Es un encabezado de salto por salto. Podemos observar que nos aparece *chunked*, esto nos dice que los datos se envían fragmentados.
  - **Vary:** Determina la forma de hacer coincidir los encabezados de las solicitudes que puedan realizarse en un futuro para decidir si se puede usar una respuesta ya almacenada en caché en lugar de una nueva desde el servidor origen.
  - **X-Content-Type-Options:** nosniff bloquea una solicitud si el tipo solicitado es *style* o *script* en algunos casos específicos.
  - **X-Frame-Options:** Usado para saber si un navegador puede ser renderizado en un <frame>, <iframe> o <object>. SAMEORIGIN nos dice que sólo la página nos puede ser mostrada en un marco del mismo origen que dicha página.
  - **X-XSS-Protection:** Característica que tiene Chrome que impide la carga de esta página cuando se detectan ciertos tipos de ataques y es obsoleta en nuevos navegadores. 1; mode=block Nos indica que se habilita el filtrado XSS para evitar que la página sea visualizada en caso de ataque.
- **Request Headers (Encabezados de Solicitud):**
  - **Accept:** Son los tipos de contenidos aceptados que el cliente puede procesar. En nuestro caso \*/\* significa que puede aceptar cualquier tipo MIME.
  - **Accept-Encoding:** Lista de codificaciones que son aceptadas; gzip, deflate, br.
  - **Accept-Language:** Lista de idiomas aceptados. Español.
  - **Authorization:** Credenciales de autorización a un usuario en el servidor, nuestra credencial es: API-KEY 60553d58c41f5dfa095b34dd
  - **Host:** Nombre del dominio o dirección IP, de uso obligatorio a partir de HTTP 1.1: api.1millionbot.com
  - **Origin:** Indica donde se origina una búsqueda. Solo incluye el nombre del servidor: <https://www.uah.es>
  - **Referer:** Contiene la dirección de la página web anterior de la que provenía el enlace a la página en la que estamos actualmente, hace referencia a la dirección web del botón atrás: <https://www.uah.es>
  - **User-Agent:** Permite identificar el protocolo de red mediante una característica que permite conocer el tipo de aplicación, sistema operativo, etc. Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/94.0.4606.81 Safari/537.36 - Nos dice que es compatible con Mozilla, el más común, después nos habla de la plataforma nativa del navegador en el que se ejecuta, en nuestro caso Windows 10 de 64 bits, después que el navegador de Chrome está basado en Gecko y por último la versión 537.36.



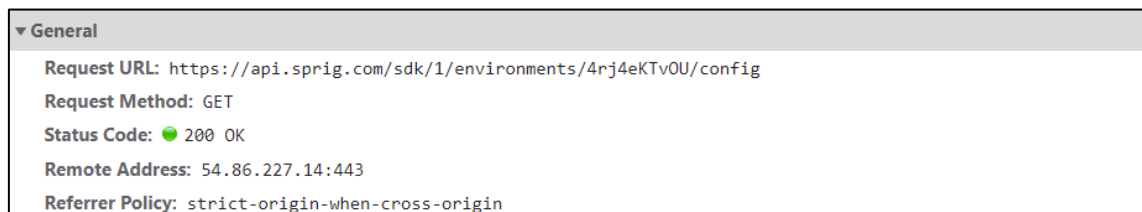
### 3.2 Cabeceras Roll20.

Primero accedemos a la página de <https://roll20.net/> y una vez dentro accedemos a las cabeceras dentro de *Network* y *Fetch/XHR*:



#### 1. Llamada “config”:

Podemos encontrar numerosas cabeceras, pero seleccionaremos la cabecera *config* al ser de las más importantes, dentro de ella podremos observar:



**Response Headers** [View source](#)

Access-Control-Allow-Origin: \*  
Connection: keep-alive  
Content-Length: 951  
Content-Type: application/json; charset=utf-8  
Date: Tue, 12 Oct 2021 11:20:46 GMT  
ETag: W/"3b7-XRrK6RFv0/NNYA1wKixgKGxKPis"  
Timing-Allow-Origin: https://0.0.com, https://0.1.com, https://1.0.com, https://1.1.com  
vary: Origin  
Via: kong/2.5.1  
X-Kong-Proxy-Latency: 0  
X-Kong-Upstream-Latency: 3  
X-Request-Id: bc9b0e4f-750d-486f-b52b-b1d8d2786a04

**Request Headers** [View source](#)

Accept: \*/\*  
Accept-Encoding: gzip, deflate, br  
Accept-Language: es-ES,es;q=0.9  
Connection: keep-alive  
Content-Type: application/json  
Host: api.sprig.com  
Origin: https://roll20.net  
Referer: https://roll20.net/  
sec-ch-ua: "Chromium";v="94", "Google Chrome";v="94", ";Not A Brand";v="99"  
sec-ch-ua-mobile: ?0  
sec-ch-ua-platform: "Windows"  
Sec-Fetch-Dest: empty  
Sec-Fetch-Mode: cors  
Sec-Fetch-Site: cross-site  
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/94.0.4606.81 Safari/537.36  
userleap-platform: web  
x-ul-sdk-version: 2.8.2

- **General:**
  - **Request-URL:** La URL de la cabecera es:  
<https://api.sprig.com/sdk/1/environments/4rj4eKTvOU/config>
  - **Request-Method:** La cabecera es de tipo GET
  - **Status-Code:** El estado de respuesta del HTTP es satisfactorio (200 OK)
  - **Remote-Address:** Su dirección remota es 54.86.227.14:443
  - **Referrer-Policy:** Determina que datos de referente, de los que se envían con cabecera *referer* han de incluirse con las solicitudes realizadas: *strict-origin-when-cross-origin* nos indica que no hay ningún referente.
- **Response Headers (Encabezados de Respuesta):**
  - **Access-Control-Allow-Origin:** \* Usamos para iniciar una petición con los servidores de respuesta.

- **Connection:** Controlamos que nuestro tipo de conexión sea *keep-alive* lo cual nos indica que se mantiene activa.
- **Content-Length:** Nos indica que el encabezado de la entidad tiene un tamaño de entidad-cuerpo de 951 bytes.
- **Content-Type:** Nos dice que el tipo de contenido que aparece en la petición de POST o PUT que será retornado al cliente. *application/json; charset=utf-8*
- **Date:** Nos proporciona tanto la fecha como la hora en la que la petición fue realizada: Tue, 12 Oct 2021 11:20:46 GMT
- **Etag:** Es un identificador para una versión específica de un recurso, esto hace que la caché sea más eficiente y que se ahorre ancho de banda en caso de que el contenido no haya cambiado. Si este cambia, el etag nos sirve como prevención de actualizaciones simultáneas de un mismo recurso.

Nuestro valor es: *W/"3b7-XRrK6RFvO/NNYAlwKixgKGxKPis"*

- **Vary:** Determina la forma de hacer coincidir encabezados de las solicitudes que puedan realizarse en un futuro para decidir si usar la almacenada en caché o utilizar una nueva desde el servidor origen: Origin
- **X-Request-Id:** Solicitud: *bc9b0e4f-750d-486f-b52b-b1d8d2786a04*
- **Request Headers (Encabezados de Solicitud):**
  - **Accept:** Son los tipos de contenidos aceptados que el cliente puede procesar. En nuestro caso *\*/\** significa que puede aceptar cualquier tipo MIME.
  - **Accept-Encoding:** Lista de codificaciones que son aceptadas; gzip, deflate, br.
  - **Accept-Language:** Lista de idiomas aceptados. Español.
  - **Content-Type:** Propiedad de cabecera utilizada para indicar el media type que será utilizado para el recurso: *application/json*
  - **Host:** Nombre del dominio o dirección IP, de uso obligatorio a partir de HTTP 1.1: *api.sprig.com*

- **Origin:** Indica donde se origina una búsqueda. Solo incluye el nombre del servidor: <https://roll20.net>
- **Referer:** Contiene la dirección de la página web anterior de la que provenía el enlace a la página en la que estamos actualmente, hace referencia a la dirección web del botón atrás: <https://roll20.net/>
- **User-Agent:** Permite identificar el protocolo de red mediante una característica que permite conocer el tipo de aplicación, sistema operativo, etc. Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/94.0.4606.81 Safari/537.36 – Es compatible con Mozilla que es el más común, usa Windows 10 de 64 bits, versión 537.36.
- **Userleap-platform:** web
- **Z-ul-sdk-version:** Nos indica el orden de un elemento posicionado y los descendientes a él. Los elementos con mayor z-index cubren a los de menor valor. Nuestro valor es: 2.8.2

### 3.3 Cabeceras El País.

Primero accedemos a las cabeceras de la página, una vez dentro seleccionamos la cabecera perteneciente al host:

The screenshot shows the Network tab of a web browser. The top request is highlighted in blue. The list of requests includes:

- config.json?key=RDFVE-2F6BM-YUS3U-DEQCL-AYV76&d=el...,Akamai,Early,EventTiming,LOGN&acao=&ak.ai=510971
- AGSKWxXINTYGI5i7aUp36nv9LWTow0Ngx9Q8G6cnQKBeyen8uC...3E262&anonid=CD14CCBB-CF4B-449C-A167-BBDF55FDEFEA
- AGSKWxXINTYGI5i7aUp36nv9LWTow0Ngx9Q8G6cnQKBeyen8uC...3E262&anonid=CD14CCBB-CF4B-449C-A167-BBDF55FDEFEA
- AGSKWxXINTYGI5i7aUp36nv9LWTow0Ngx9Q8G6cnQKBeyen8uC...3E262&anonid=CD14CCBB-CF4B-449C-A167-BBDF55FDEFEA
- AGSKWxVfuoQ0fqa4Yg0b1GtVemYC7OZ3qf8\_Xfyyle9JTLxGks...lfeVkbNKKZHKsjY2GMNwNIV6pXqvn2lkOdqxMId0JZ-KA6w==
- AGSKWxVfuoQ0fqa4Yg0b1GtVemYC7OZ3qf8\_Xfyyle9JTLxGks...lfeVkbNKKZHKsjY2GMNwNIV6pXqvn2lkOdqxMId0JZ-KA6w==
- ?host=elpais.com&domain=elpais.com&path=%2F
- AGSKWxUd9QLI\_wy2POLqvs-saCkXX\_p7SN4-30e9b7RuOE5ivV...dvsIY-CDWbRE9bYgZB2dBDTsrPT-LRKumkU9Pv\_4zmb4zA==
- AGSKWxUd9QLI\_wy2POLqvs-saCkXX\_p7SN4-30e9b7RuOE5ivV...dvsIY-CDWbRE9bYgZB2dBDTsrPT-LRKumkU9Pv\_4zmb4zA==
- AGSKWxUd9QLI\_wy2POLqvs-saCkXX\_p7SN4-30e9b7RuOE5ivV...dvsIY-CDWbRE9bYgZB2dBDTsrPT-LRKumkU9Pv\_4zmb4zA==
- AGSKWxUcAMShHIKkhrxAu81wl0iV\_maG5L8dJn75q6T2DEJ53C...Al4JcKdS4RvVAhyHp0SIP5a5500RJkilo-59I3FrTpfDQ==
- AGSKWxUcAMShHIKkhrxAu81wl0iV\_maG5L8dJn75q6T2DEJ53C...Al4JcKdS4RvVAhyHp0SIP5a5500RJkilo-59I3FrTpfDQ==
- AGSKWxUcAMShHIKkhrxAu81wl0iV\_maG5L8dJn75q6T2DEJ53C...Al4JcKdS4RvVAhyHp0SIP5a5500RJkilo-59I3FrTpfDQ==
- config.json?key=RDFVE-2F6BM-YUS3U-DEQCL-AYV76&d=el...cn=%2F%2F364bf6cc.akstat.io%2F&acao=&ak.ai=510971

Una vez dentro procederemos a ver su contenido:

▼ General
Request URL: https://mab.chartbeat.com/mab_strategy/headline_testing/get_strategy/?host=elpais.com&domain=elpais.com&path=%2F
Request Method: GET
Status Code: 200
Remote Address: 151.101.134.202:443
Referrer Policy: strict-origin-when-cross-origin

▼ Response Headers
accept-ranges: bytes
access-control-allow-origin: *
age: 27
cache-control: no-store, no-cache, must-revalidate, max-age=0, s-maxage=0
content-encoding: gzip
content-length: 598
content-type: application/json
cross-origin-resource-policy: cross-origin
date: Wed, 13 Oct 2021 17:15:06 GMT
expires: Mon, 11 Oct 2021 17:14:39 GMT
vary: Origin, Access-Control-Request-Headers, Access-Control-Request-Method, Accept-Encoding
via: 1.1 varnish (Varnish/6.0), 1.1 varnish
x-cache: HIT
x-cache-hits: 18
x-served-by: cache-mad22047-MAD
x-timer: S1634145306.444814,VS0,VE0

▼ Request Headers
:authority: mab.chartbeat.com
:method: GET
:path: /mab_strategy/headline_testing/get_strategy/?host=elpais.com&domain=elpais.com&path=%2F
:scheme: https
accept: */*
accept-encoding: gzip, deflate, br
accept-language: es-ES,es;q=0.9
origin: https://elpais.com
referer: https://elpais.com/
sec-ch-ua: "Chromium";v="94", "Google Chrome";v="94", ";Not A Brand";v="99"
sec-ch-ua-mobile: ?0
sec-ch-ua-platform: "Windows"
sec-fetch-dest: empty
sec-fetch-mode: cors
sec-fetch-site: cross-site
user-agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/94.0.4606.81 Safari/537.36

▼ Query String Parameters	<a href="#">view source</a>	<a href="#">view URL-encoded</a>
host: elpais.com		
domain: elpais.com		
path: /		

Tenemos:

- **General:**
  - **Request-URL:** La URL de la cabecera es:  
[https://mab.chartbeat.com/mab\\_strategy/headline\\_testing/get\\_strategy/?host=elpais.com&domain=elpais.com&path=%2F](https://mab.chartbeat.com/mab_strategy/headline_testing/get_strategy/?host=elpais.com&domain=elpais.com&path=%2F)
  - **Request-Method:** La cabecera es de tipo GET
  - **Status-Code:** El estado de respuesta del HTTP es satisfactorio (200)
  - **Remote-Address:** Su dirección remota es 151.101.134.202:443
  - **Referrer-Policy:** Determina que datos de referente, de los que se envían con cabecera *referer* han de incluirse con las solicitudes realizadas: *strict-origin-when-cross-origin* nos indica que no hay ningún referente
- **Response Headers:**
  - **Accept-ranges:** El marcador usado por el servidor que notifica que solicitudes parciales soporta, es medida en bytes.
  - **Access-control-allow-origin:** \* Inicia una petición para los servidores con respuesta, usando la cabecera ya mencionada.
  - **Age:** Contiene el tiempo que el objeto ha estado almacenado en memoria caché, en nuestro caso son 27 segundos.
  - **Cache-control:** Las políticas de nuestro caché son:
    - **No-Store:** La respuesta puede que no se almacene en cualquier caché.
    - **No-Cache:** La respuesta puede estar almacenada por cualquier caché, incluso cuando esta petición no es cacheable
    - **Must-Revalidate:** Cada vez que un recurso pasa a estar obsoleto, el cache no ha de usar esta copia obsoleta sin ser validada anteriormente por el servidor en el que se ha originado.
  - **content-encoding:** Usada para comprimir el media-type y permite decodificarlo al cliente. En nuestro caso tenemos gzip, originalmente es el formato de UNIX y es usado por Lempel-Ziv coding con un CRC de 32 bits.
  - **content-length:** El tamaño entidad cuerpo es 598 bytes.

- **content-type:** Nos dice el tipo de contenido que aparece en la petición de POST o PUT que será retornado al cliente. *application/json*.
- **cross-origin-resource-policy:** Es la respuesta del encabezado HTTP para bloquear peticiones *cross-origin*.
- **date:** Nos proporciona la fecha y la hora a la que hemos realizado la petición. En nuestro caso: Wed, 13 Oct 2021 17:30:28 GMT
- **expires:** Nos menciona tanto la hora como la fecha en donde la respuesta será considerada como caducada: Mon, 11 Oct 2021 17:30:27 GMT
- **vary:** Determina la forma de hacer coincidir los encabezados de las solicitudes que puedan realizarse en un futuro para decidir si se puede usar una respuesta ya almacenada en caché en lugar de una nueva desde el servidor origen: Origin, Access-Control-Request-Headers, Access-Control-Request-Method, Accept-Encoding
- **Request Headers:**
  - **accept:** Son los tipos de contenidos aceptados que el cliente puede procesar. En nuestro caso \*/\* significa que puede aceptar cualquier tipo MIME.
  - **accept-encoding:** Lista de codificaciones que son aceptadas; gzip, deflate, br.
  - **accept-language:** Lista de idiomas aceptados. Español.
  - **origin:** Indica donde se origina una búsqueda. Solo incluye el nombre del servidor: <https://elpais.com>
  - **referer:** Contiene la dirección de la página web anterior de la que provenía el enlace a la página en la que estamos actualmente, hace referencia a la dirección web del botón atrás: <https://elpais.com/>
  - **user-agent:** Permite identificar el protocolo de red mediante una característica que permite conocer el tipo de aplicación, sistema operativo, etc. *Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/94.0.4606.81 Safari/537.36* - Nos dice que es compatible con Mozilla, el más común, después nos habla de la plataforma nativa del navegador en el que se ejecuta, en nuestro caso Windows 10 de 64 bits, después que el

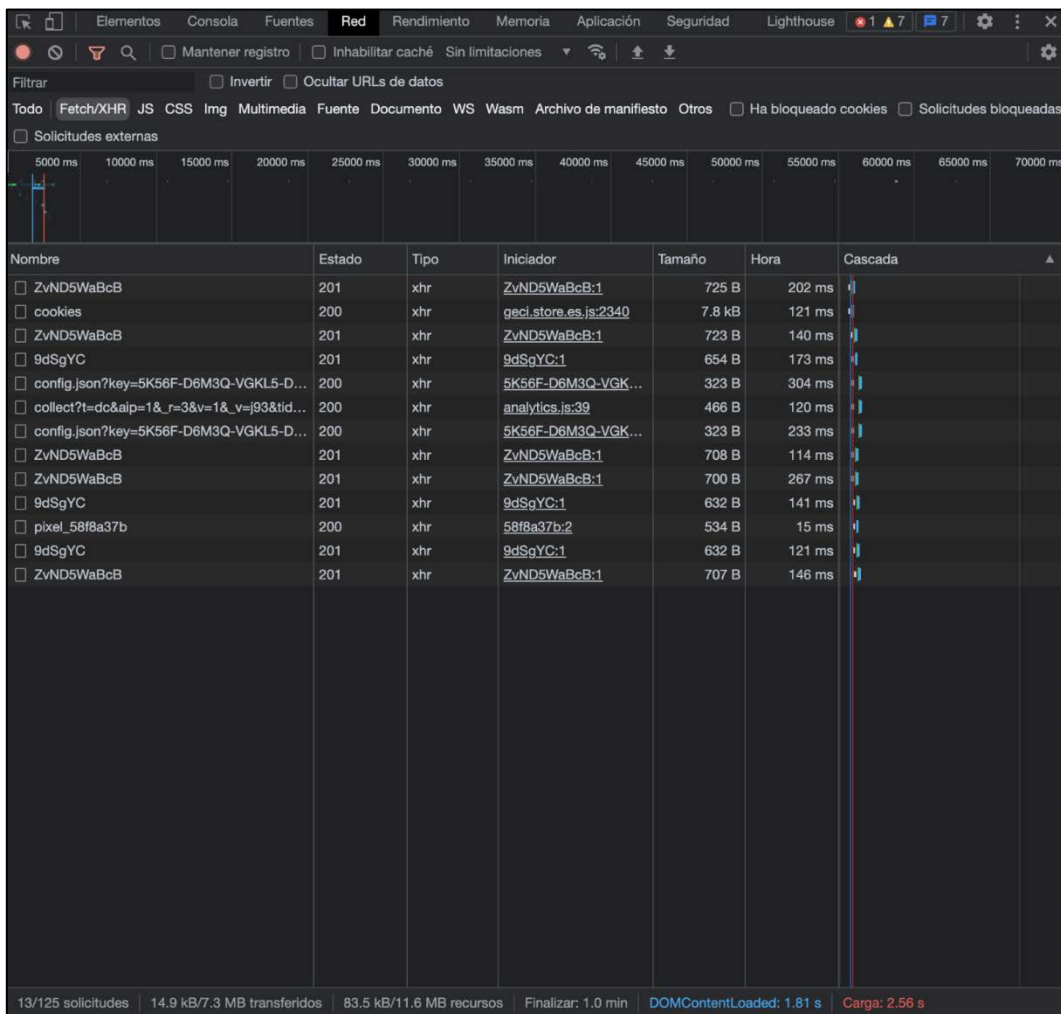


navegador de Chrome está basado en Gecko y por último la versión 537.36.

- **Query String Headers:**
  - *host*: El host es: elpais.com
  - *domain*: El dominio es: elpais.com

### 3.4 Cabeceras El Corte Inglés.

Al acceder a la [URL](#) haremos el procedimiento explicado al principio de este punto para ver las cabeceras que nos interesan, podemos ver las siguientes:



Nombre	Estado	Tipo	Iniciador	Tamaño	Hora	Cascada
<input type="checkbox"/> ZvND5WaBcB	201	xhr	ZvND5WaBcB:1	725 B	202 ms	
<input type="checkbox"/> cookies	200	xhr	geci.store.es.js:2340	7.8 kB	121 ms	
<input type="checkbox"/> ZvND5WaBcB	201	xhr	ZvND5WaBcB:1	723 B	140 ms	
<input type="checkbox"/> 9dSgYC	201	xhr	9dSgYC:1	654 B	173 ms	
<input type="checkbox"/> config.json?key=5K56F-D6M3Q-VGKL5-D...	200	xhr	5K56F-D6M3Q-VGK...	323 B	304 ms	
<input type="checkbox"/> collect?t=dc&aip=1&_r=3&v=1&_v=j93&tid...	200	xhr	analytics.js:39	466 B	120 ms	
<input type="checkbox"/> config.json?key=5K56F-D6M3Q-VGKL5-D...	200	xhr	5K56F-D6M3Q-VGK...	323 B	233 ms	
<input type="checkbox"/> ZvND5WaBcB	201	xhr	ZvND5WaBcB:1	708 B	114 ms	
<input type="checkbox"/> ZvND5WaBcB	201	xhr	ZvND5WaBcB:1	700 B	267 ms	
<input type="checkbox"/> 9dSgYC	201	xhr	9dSgYC:1	632 B	141 ms	
<input type="checkbox"/> pixel_58f8a37b	200	xhr	58f8a37b:2	534 B	15 ms	
<input type="checkbox"/> 9dSgYC	201	xhr	9dSgYC:1	632 B	121 ms	
<input type="checkbox"/> ZvND5WaBcB	201	xhr	ZvND5WaBcB:1	707 B	146 ms	

13/125 solicitudes | 14.9 kB/7.3 MB transferidos | 83.5 kB/11.6 MB recursos | Finalizar: 1.0 min | DOMContentLoaded: 1.81 s | Carga: 2.56 s

Ahora procederemos al análisis de cada una de ellas en particular. Podemos observar que hay muchas de ellas que son iguales, estas se utilizan principalmente para la actividad de la página. Si empezamos a detallarlas vemos lo siguiente:



## Llamada “ZvND5WaBcB”

```

▼ General
Solicitar URL: https://www.elcorteingles.es/8GK70azSVoIP9DC7yyyc7Umm/S7J5fL0JLY/K1JQKC4AQg/P3/ZvND5WaBcB
Método de la solicitud: POST
Código de estado: 201
Dirección remota: 2.17.153.31:443
Política de referencia: strict-origin-when-cross-origin

▼ Encabezados de respuesta
access-control-allow-credentials: true
access-control-allow-headers: Content-Type
access-control-allow-origin: https://www.elcorteingles.es
content-type: application/json
date: Sat, 09 Oct 2021 11:59:07 GMT
server-timing: edge; dur=22
server-timing: origin; dur=119
server-timing: cdn-cache; desc=MISS
set-cookie: _abck=C73F0822AB1F439488532CF41ECE7693~1~YAAQfNMRAkB/HLJ8AQAA3MzrZAao75j7dYhkj5nvFi0c0a73nnpnFIsERkGi9TtN5r5Lc
vary: Origin

```

```

▼ Encabezados de solicitud
:authority: www.elcorteingles.es
:method: POST
:path: /8GK70azSVoIP9DC7yyyc7Umm/S7J5fL0JLY/K1JQKC4AQg/P3/ZvND5WaBcB
:scheme: https
accept: */*
accept-encoding: gzip, deflate, br
accept-language: es-ES,es;q=0.9
content-length: 1504
content-type: text/plain;charset=UTF-8
cookie: SSLB=1; SSID=CAAVLB00AAAAAAJhGFhSWVCCgmEYWEBAAAAAACYRhyQD53Jz6AAG4BSIACYRhyQEA; SSSC=567.G7017034868781638
kd4fdabqvl&ss=kujqvg7h&sl=0&tt=0"; SSPV=LKYAAAAAAGAAAAAAMAAAAA; bm_mi=31D2F5B50E4F45FB4A417DDB117FCB45
origin: https://www.elcorteingles.es
referer: https://www.elcorteingles.es/
sec-ch-ua: "Chromium";v="94", "Google Chrome";v="94", ";Not A Brand";v="99"
sec-ch-ua-mobile: ?0
sec-ch-ua-platform: "macOS"
sec-fetch-dest: empty
sec-fetch-mode: cors
sec-fetch-site: same-origin
user-agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10_15_7) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/94.0.4606.71 Sa

```

Dentro de los parámetros generales vemos:

- **Solicitar URL**, es la URL a la que se solicita la información en el backend
- **Método de solicitud**, se trata de una solicitud POST.
- **Status Code**, nos devuelve un código 2XX indicando que se ha procesado la solicitud y está todo OK, en concreto el código 201, nos dice que se han creado nuevos elementos o recursos.
- **Política de referencia**, nos indica que es *strict-origin-when-cross-origin*, esto es que cuando el hipervínculo es dentro del mismo dominio, se envía la URL completa, y cuando el hipervínculo es a otro origen, se envía entonces solo el dominio y el protocolo de la URL, sin el path.

Luego si vamos a los encabezados de respuesta encontramos la siguiente información:

- ***access-control-allow-credentials***, esta cabecera nos indica que el código debe exponerse a los ficheros JS, en caso de que no debieran, la cabecera no existiría.
- ***access-control-allow-headers***, nos indica que solo los encabezados que sean *Content-Type* pueden ser usados en la solicitud, si nos vamos más abajo podemos ver que se podrá usar contenido *application/json*, esto se puede ver entre los encabezados de la solicitud.
- ***access-control-allow-origin***, nos indica que los recursos de código solo pueden provenir de las páginas que tengan el mismo *access-control-allow-origin*, es decir los que provengan de <https://elcorteingles.es>
- ***content-type***, como se ha mencionado en el párrafo de arriba nos está devolviendo un contenido de tipo *application/json*.
- ***date***, nos indica la fecha en la que se ha respondido la petición.
- ***server-timing***, esta cabecera se devuelve por triplicado, nos indica métricas del servidor, permitiendo saber que hasta la primera conexión con el servidor se tardan 22ms, que el tiempo que tarda al origen es de 119ms, e indicando que no se ha podido cargar nada de la caché.
- ***set-cookie***, nos indica las cookies que nos asignan automáticamente, además si nos vamos al final de las definiciones podemos ver otros parámetros interesantes de estas cookies:

```
Domain=.elcorteingles.es; Path=/; Expires=Sun, 09 Oct 2022 15:24:50 GMT; Max-Age=31536000; Secure
```

Esto nos especifica que el dominio donde se tiene que enviar la cookie es el propio de *El Corte Ingles*, así como nos especifica un path que es donde la cookie debe ser enviada, así como un campo *Expires* y otro *Max-Age*, lo que nos indica hasta cuando se guarda esta cookie en nuestro dispositivo, al estar especificados los dos el campo de *Max-Age* tiene preferencia, también tenemos el parámetro *Secure*, que nos indica que la cookie solo se envía al servidor cuando se hace una petición *HTTPS*.

- ***vary***, esta cabecera nos indica como relacionar futuras peticiones, en concreto cuando aparece el valor de *Origin* este es generado dinámicamente por la petición de *CORS*.

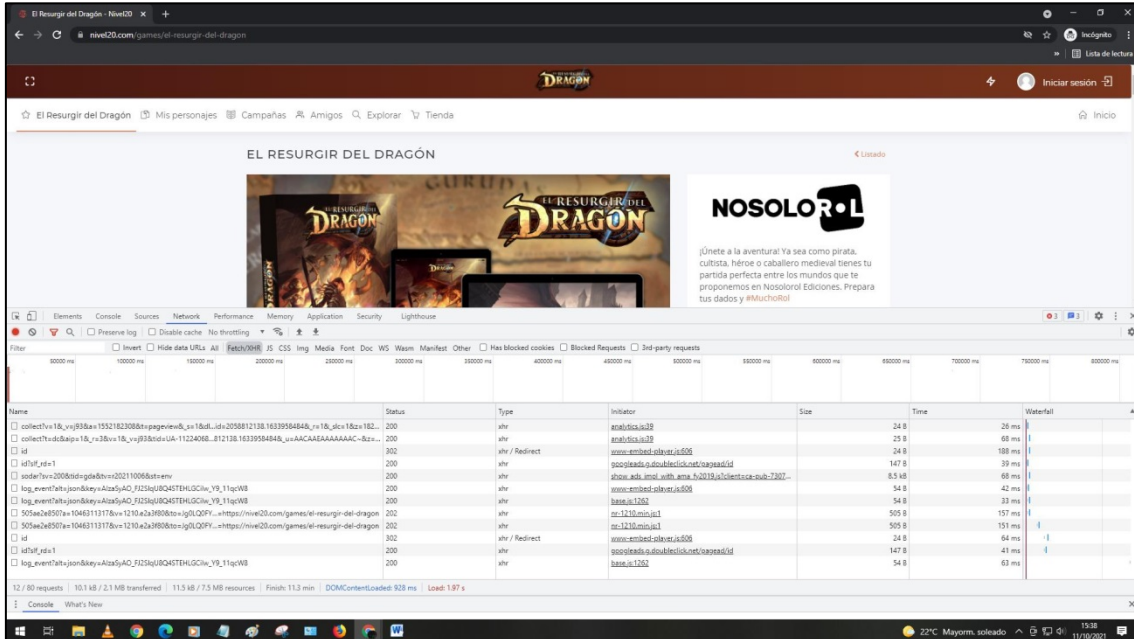
Si inspeccionamos ahora la parte de los encabezados de solicitud podemos ver los siguientes:

- ***:authority***, se trata de un pseudo-encabezado nuevo en *HTTP/2*, se corresponde con el antiguo *host*, y nos indica la autoridad URI del destino, es decir porción de información del usuario.
- ***:method***, es un pseudo-encabezado, que nos indica el método de la petición.
- ***:path***, envía el path de la solicitud *HTTP*.
- ***:scheme***, indica el método de acceso de la solicitud, en este caso es *https*.
- ***accept***, indica el tipo de contenido que el cliente está dispuesto a aceptar, en este caso es cualquier tipo de contenido *MIME* (*Multipurpose Internet Mail Extensions*).

- ***accept-encoding***, nos indica los tipos de codificación que nuestra máquina entiende, y por lo tanto en los que se tiene que mandar el contenido.
- ***accept-language***, mandamos que preferimos el contenido en español, adicionalmente indicamos que la prioridad es 0.8 mediante el parámetro *q*.
- ***content-length***, especificamos la longitud en bytes del contenido que vamos a mandar.
- ***content-type***, el contenido que vamos a mandar es de tipo texto, así mismo está con los caracteres contenido en UTF-8.
- ***cookie***, es una lista de las cookies que estamos mandando junto con la solicitud
- ***origin***, es la dirección que indica de el origen de donde se está haciendo la solicitud.
- ***referer***, es la dirección desde donde se está haciendo la solicitud
- ***sec-ch-ua***, da información al servidor de la máquina que está solicitando la información.
- ***sec-ch-ua-mobile***, da información del dispositivo móvil desde donde se solicita la información, en este caso tiene un valor ?0 porque lo estamos haciendo desde un ordenador.
- ***sec-ch-ua-platform***, da información sobre la plataforma desde donde se hace la solicitud.
- ***sec-fetch-dest***, indica el destino de la información solicitada.
- ***sec-fetch-mode***, indica el modo en el que se hace la solicitud, en este caso sigue el protocolo *cors*.
- ***sec-fetch-site***, indica la relación entre el sitio desde donde se requiere la información y desde donde se recupera, en este caso el *same-site* indicando que va a ser destinada a la misma página.
- ***user-agent***, nos permite identificar el protocolo de red que ayuda a descubrir el tipo de aplicación, sistema operativo, proveedor del software o la versión del software de la petición del agente de usuario.

### 3.5 Cabeceras nivel20.

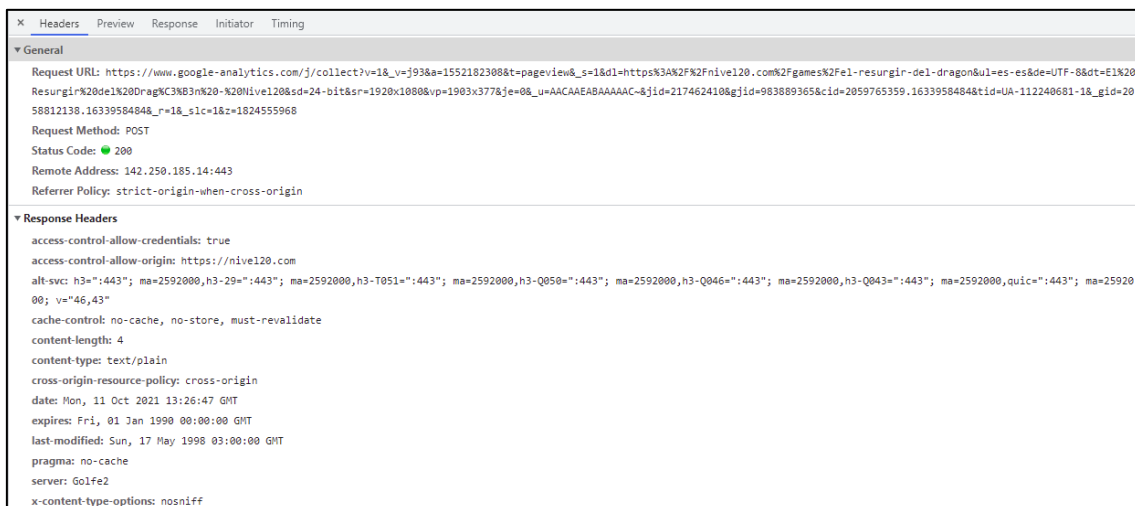
Primero accedemos a la página de <https://nivel20.com/games/el-resurgir-del-dragon> y una vez dentro accedemos a las cabeceras dentro de *Network* y *Fetch/XHR*:



Name	Status	Type	Initiator	Size	Time	Waterfall
collect?v=1&v=j93&a=1552182308&t=pageview&s=1&id=2058812138.1633958484&r=1&slc=1&z=1824555968	200	xhr	analytics.js	24 B	26 ms	
collect?t=dc&aip=1&r=3&v=1&v=j93&tid=UA-11224068-812138.1633958484&u=AAACAAEAAAAAAC~&z=1173543143	200	xhr	analytics.js	25 B	68 ms	
id	302	xhr / Redirect	www.embed-silver.es/806	24 B	58 ms	
id?slf_rd=1	200	xhr	googleads.g.doubleclick.net/pagead/clk	147 B	39 ms	
sodar?sv=200&tid=gda&tv=r20211006&st=env	200	xhr	show.ads.msl.with.ama.5.2018.01clientica-sub-7207-	8.5 kB	68 ms	
log_event?alt=json&key=AlzaSyAO_FJ2SlqU8Q4STEHLGCilw_Y9_11qcW8	200	xhr	www.embed-silver.es/806	34 B	42 ms	
log_event?alt=json&key=AlzaSyAO_FJ2SlqU8Q4STEHLGCilw_Y9_11qcW8	200	xhr	base.u.1262	34 B	33 ms	
505ae2e850?a=1046311317&v=1210.e2a3f80&to=Jg0LQ0FY...=https://nivel20.com/games/el-resurgir-del-dragon	202	xhr	rs-1210.mosai1	505 B	157 ms	
505ae2e850?a=1046311317&v=1210.e2a3f80&to=Jg0LQ0FY...=https://nivel20.com/games/el-resurgir-del-dragon	202	xhr	rs-1210.mosai1	505 B	151 ms	
id	302	xhr / Redirect	www.embed-silver.es/806	24 B	64 ms	
id?slf_rd=1	200	xhr	googleads.g.doubleclick.net/pagead/clk	147 B	41 ms	
log_event?alt=json&key=AlzaSyAO_FJ2SlqU8Q4STEHLGCilw_Y9_11qcW8	200	xhr	base.u.1262	34 B	63 ms	

Name
collect?v=1&v=j93&a=1552182308&t=pageview&s=1&id=2058812138.1633958484&r=1&slc=1&z=1824555968
collect?t=dc&aip=1&r=3&v=1&v=j93&tid=UA-11224068-812138.1633958484&u=AAACAAEAAAAAAC~&z=1173543143
id
id?slf_rd=1
sodar?sv=200&tid=gda&tv=r20211006&st=env
log_event?alt=json&key=AlzaSyAO_FJ2SlqU8Q4STEHLGCilw_Y9_11qcW8
log_event?alt=json&key=AlzaSyAO_FJ2SlqU8Q4STEHLGCilw_Y9_11qcW8
505ae2e850?a=1046311317&v=1210.e2a3f80&to=Jg0LQ0FY...=https://nivel20.com/games/el-resurgir-del-dragon
505ae2e850?a=1046311317&v=1210.e2a3f80&to=Jg0LQ0FY...=https://nivel20.com/games/el-resurgir-del-dragon
id
id?slf_rd=1
log_event?alt=json&key=AlzaSyAO_FJ2SlqU8Q4STEHLGCilw_Y9_11qcW8
log_event?alt=json&key=AlzaSyAO_FJ2SlqU8Q4STEHLGCilw_Y9_11qcW8
log_event?alt=json&key=AlzaSyAO_FJ2SlqU8Q4STEHLGCilw_Y9_11qcW8
log_event?alt=json&key=AlzaSyAO_FJ2SlqU8Q4STEHLGCilw_Y9_11qcW8

Ahora procedemos al análisis de una de las cabeceras de tipo POST que, en este caso, es la primera de la lista.



Dentro de los parámetros generales vemos:

- **Request URL**, es la URL a la que se solicita la información.
- **Request Method**, se trata de una solicitud POST.
- **Status Code**, nos devuelve un código 2XX indicando que se ha procesado la solicitud y está todo *OK*, en concreto el código 200, es la respuesta estándar para las peticiones correctas.
- **Referrer Policy**, nos indica que es *strict-origin-when-cross-origin* y, como ya hemos visto anteriormente, significa que cuando el hipervínculo está dentro del mismo dominio, se envía la URL completa, y cuando el hipervínculo es a otro origen, se envía entonces solo el dominio y el protocolo de la URL, sin el path.

A continuación tenemos los encabezados de respuesta, donde encontramos la siguiente información:

- **access-control-allow-credentials**, esta cabecera nos indica que el código debe exponerse a los ficheros JS que, en caso de no exponerse, la cabecera no existiría.
- **access-control-allow-origin**, nos indica que los recursos de código solo pueden provenir de las páginas que tengan el mismo *access-control-allow-origin*, es decir, los que provengan de <https://nivel20.com>
- **alt-svc**, indica que un recurso en particular debe cargarse desde otro servidor, aunque a nosotros como usuarios nos parece que se carga desde el mismo.
- **cache-control**, contiene las instrucciones para almacenar en caché tanto las solicitudes como las respuestas. En este caso, la respuesta no debe guardarse ni almacenarse y debe validarse obligatoriamente.
- **content-length**, indica el tamaño del mensaje al destinatario en bytes.
- **content-type**, indica el tipo de medio original del recurso que, en nuestro caso, es texto plano.

- ***cross-origin-resource-policy***, indica si el navegador debe bloquear las solicitudes de origen cruzado al recurso dado.
- ***date***, indica la fecha de la consulta.
- ***expires***, indica la fecha a la que el mensaje se considerará caducado.
- ***last-modified***, contiene la fecha de la última modificación del servidor de origen.
- ***pragma***, este encabezado sirve para la compatibilidad con versiones anteriores de la caché que no tienen un control de las mismas. En nuestro caso, obliga a la caché a enviar la solicitud al servidor de origen para su validación antes de que se publique una copia en la propia caché.
- ***server***, describe el software utilizado por el servidor de origen que manejó la solicitud, es decir, el servidor que generó la respuesta.
- ***x-content-type-options***, es un marcador utilizado por el servidor para indicar que los tipos MIME anunciados en los encabezados *content-type* deben seguirse y no cambiarse.

Por último, tenemos los encabezados de solicitud, donde nos encontramos la siguiente información:

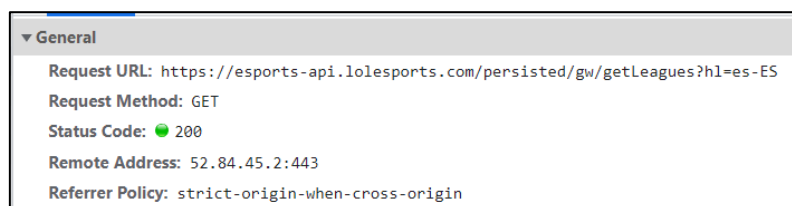
- ***:authority:***, como ya hemos visto anteriormente, se trata de un pseudo-encabezado nuevo en *HTTP/2*, que se corresponde con el antiguo host, y nos indica la autoridad URI del destino, es decir, la porción de información del usuario.
- ***:method:***, también se trata de un pseudo-encabezado que nos indica el método de la petición.
- ***:path:***, envía la ruta de la solicitud *HTTP*.
- ***:scheme:***, indica el método de acceso de la solicitud.
- ***accept***, indica el tipo de contenido que el cliente está dispuesto a aceptar, en este caso es cualquier tipo de contenido *MIME* (*Multipurpose Internet Mail Extensions*).
- ***accept-encoding***, nos indica los tipos de codificación que nuestra máquina entiende y en los que se tiene que mandar el contenido.
- ***accept-language***, mandamos que preferimos el contenido en español, adicionalmente indicamos que la prioridad es 0.9 mediante el parámetro *q*.
- ***content-length***, especificamos la longitud en bytes del contenido que vamos a mandar.
- ***content-type***, como ya hemos visto anteriormente, indica el tipo de medio original del recurso que, en nuestro caso, es texto plano.
- ***origin***, indica el punto de origen de la petición de recogida que, en nuestro caso, es <https://nivel20.com>
- ***referer***, indica la dirección de la página web previa desde la cual un link nos ha redirigido a la actual, <https://nivel20.com>
- ***sec-ch-ua***, da información al servidor de la máquina que está solicitando la información.
- ***sec-ch-ua-mobile***, da información del dispositivo móvil desde donde se solicita la información que, en nuestro caso, tiene un valor “?0” porque lo estamos haciendo desde un ordenador.



- **sec-ch-ua-platform**, indica la plataforma desde donde se está realizando la solicitud.
- **sec-fetch-dest**, indica el destino de la información solicitada.
- **sec-fetch-mode**, indica el modo en el que se hace la solicitud, en este caso sigue el protocolo *cors*.
- **sec-fetch-site**, indica la relación entre el sitio desde donde se requiere la información y desde donde se recupera, en este caso el *same-site* indicando que va a ser destinada a la misma página.
- **user-agent**, nos permite identificar el protocolo de red que ayuda a descubrir el tipo de aplicación, sistema operativo, proveedor del software o la versión del software de la petición del agente de usuario.

### 3.6 Cabeceras lolesports.

Por último, vamos a analizar las cabeceras de una llamada GET de la página de lolesports, en concreto la llamada es getLeague.



Podemos diferenciar tres categorías de cabeceras, las generales, las response y las request, las cuales veremos a continuación:



```
▼ Request Headers
:authority: esports-api.lolesports.com
:method: GET
:path: /persisted/gw/getLeagues?hl=es-ES
:scheme: https
accept: */*
accept-encoding: gzip, deflate, br
accept-language: es-419,es;q=0.9,en;q=0.8
origin: https://lolesports.com
referer: https://lolesports.com/
sec-ch-ua: "Chromium";v="94", "Google Chrome";v="94", ";Not A Brand";v="99"
sec-ch-ua-mobile: ?0
sec-ch-ua-platform: "Windows"
sec-fetch-dest: empty
sec-fetch-mode: cors
sec-fetch-site: same-site
user-agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/94.0.4606.81 Safari/537.36
x-api-key: 0TvQnueqKa5mxJntVWt0w4LpLfEkrV1Ta8rQBb9Z
```

Dentro de cada categoría observamos los siguientes parámetros:

- **General:**

- **Request-URL:**  
https://esports-api.lolesports.com/persisted/gw/getLeagues?hl=es-ES
- **Request-Method:** La cabecera es de tipo GET.
- **Status-Code:** El estado de respuesta del HTTP es satisfactorio (200).
- **Remote-Address:** Su dirección remota es 52.84.45.2:443
- **Referrer-Policy:** Determina los datos del referente, que se envían con cabecera referer habrán de incluirse con las solicitudes realizadas: strict-origin-when-cross-origin, que nos indica que no hay ningún referente.

- **Response Headers:**

- **Access-control-allow-headers:** Es una respuesta a una solicitud para saber que encabezados pueden ser utilizados durante la propia solicitud.
- **access-control-allow-methods:** Es igual a la cabecera anterior, pero en vez de ser las cabeceras son los métodos.
- **access-control-allow-origin:** Inicia una petición a los servidores, usando una cabecera anterior.
- **age:** Contiene el tiempo que el objeto ha estado almacenado en la caché, en este caso 4 segundos.
- **Cache-control:** Controla el tiempo máximo en la que algo puede estar en la caché, en este caso, 60 segundos.



- **Content-encoding:** Esta cabecera es usada para comprimir el contenido, es decir el tipo de media que se use. En este caso gzip.
  - **Content-length:** Esta cabecera es usada para determinar el tamaño del contenido. En este caso 1804 bytes.
  - **Content-type:** Esta cabecera es usada para determinar el tipo del contenido, en nuestro caso es una aplicación/json.
  - **Date:** Determina la fecha.
  - **Via:** Esta cabecera nos muestra la forma en la que la información llega a nosotros, en este caso a través de dos cloudfont.
  - **X-amz-apigw-id:** Cabecera de comportamiento y solicitudes de conexión de clouz front.
  - **X-amz-cf-id:** Cabecera de comportamiento y solicitudes de conexión de clouz front.
  - **X-amz-cf-pop:** Cabecera de comportamiento y solicitudes de conexión de clouz front. Que determina dónde tiene que salir la información. Esta cabecera está duplicada, ya que sale por dos sitios distintos.
  - **X-amzn-requested:** Cabecera de comportamiento y solicitudes de conexión de clouz front, que determina la solicitud en sí misma.
  - **X-amzn-trace-id:** Cabecera de comportamiento y solicitudes de conexión de clouz front que determina la ID root que inicia la solicitud.
  - **X-cache:** Esta cabecera determina dónde se está usando la caché.
- **Request Headers:**
    - **:authority:** Cabecera que determina un pseudo encabezado de la página, en este caso: esports-apo.lolesports.com
    - **:method:** Este es otro pseudo encabezado que nos dice el tipo de método, en este caso GET.
    - **:path:** Esta cabecera envía la ruta de la solicitud HTTP que se pide. Esta es /persisted/gw/getLeague?hl=es-ES
    - **:scheme:** Cabecera que determina el tipo de acceso a la solicitud, en este caso http.

- **Accept:** Cabecera que muestra los tipos de contenidos que el cliente acepta procesar. En este caso, todos.
- **Accept-encoding:** Cabecera que muestra los tipos de codificación que el cliente acepta procesar. En este caso, gzip, deflate y br.
- **Accept language:** Cabecera que muestra los tipos de idiomas que el cliente acepta. En este caso, es-419, es; q=0.9. (Que corresponde al español) y en; q=0.8. (Que corresponde al inglés)
- **Origin:** Cabecera que indica dónde se da la búsqueda. En nuestro caso en <https://lolesports.com>
- **Referer:** Cabecera que contiene la página web anterior de la que venimos. En este caso <https://lolesports.com/>
- **Sec-ch-ua:** Cabecera que da la información al servidor sobre el ordenador que hace la llamada. En este caso: “Chromium”; v=”94”, “Google chrome”; v=”94”, “; Not A Brand”; v=”99”.
- **Sec-ch-ua-mobile:** Cabecera que da la información al servidor sobre el móvil que hace la llamada. En este caso: ?0 es decir, no hay móvil.
- **Sec-ch-ua-platform:** Cabecera que da la información al servidor sobre el sistema operativo que hace la llamada. En este caso Windows.
- **Sec-fech-dest:** Cabecera el destino de la información solicitada. En este caso vacío, ya que no hay destino.
- **Sec-fech-mode:** Cabecera que indica el protocolo en el que está la solicitud, en este caso es el cors.
- **Sec-fech-site:** Cabecera que establece la relación entre quien hace la llamada y a dónde llega. En este caso es el mismo sitio.
- **User-agent:** Cabecera que permite identificar el protocolo de red que permite reconocer una serie de características. En este caso: Mozilla/5.0 (Windows NT 10.0; WIN64 ; x64) AppleWebKit/537.36(KHTML, like Gecko) Chrome/94.04606.81 Safari/537.36
- **X-api-key:** Cabecera que da la autorización para todas las claves API.

#### **4. Consideraciones finales.**

A través de esta práctica hemos podido conocer y utilizar las herramientas de desarrollador que nos proporcionan nuestros navegadores web. Hemos visto de primera mano cómo trabajan las diferentes peticiones y respuestas HTTP que se realizan al cliente o al servidor y observar qué información es la que se está utilizando en cada transacción a la vez que aprendíamos qué es lo que hace cada una de ellas.