



Programación en la Web

Taller No. 1

Estudiantes: Cristhian Rafael Lizcano Arenas
– Duvan Ferney Vargas

Código: 2191050 - 2191120

Grupo:J1

Escuela de ingeniería de sistemas e informática
Universidad Industrial de Santander

12 de Marzo de 2022

Taller 1:

1. Acceder a un sitio estático usando un navegador

- Mirar headers e info interesante

[Categoría:Personajes](#) | [My Hero Academia Wiki](#) | [Fandom](#)

The image shows a screenshot of the 'Personajes' (Characters) category page on the My Hero Academia Wiki Fandom. The page features a grid of character images including Katsuki Bakugo, Toya Todoroki, Tomura Shigaraki, Himiko Toga, Shoto Todoroki, Keigo Takami, Shota Aizawa, and Ochaco Uraraka. To the right is a large promotional banner for 'The Matrix Resurrections' on HBO Max, advertising a 33% discount. Below the character grid, it says 'Todos los artículos (260)'. At the bottom, there are links to 'Follow on IG', 'Newsletter', and 'Join Fan Lab'. A yellow banner at the very bottom reads 'Bingebot: Find a new show to watch in 30 seconds'.

Below the webpage screenshot, the network headers for the URL `https://myheroacademia.fandom.com/wiki/Personajes` are displayed in the Insomnia browser. The headers are as follows:

NAME	VALUE
content-type	text/html; charset=UTF-8
x-content-type-options	nosniff
content-language	es
content-security-policy	upgrade-insecure-requests
content-security-policy-report-only	script-src 'unsafe-eval' 'self' https: 'self' data: 'unsafe-inline' 'unsafe-eval' blob: 'unsafe-inline' internal-soap.wikia.com internal-soap.fandom.com internal-soap.wikia.org internal-soap.gamepedia.com www.fandom.com www.wikia.com www.wikia.org www.gamepedia.com; default-src 'self' data: https://images.wikia.com https://static.wikia.nocookie.net https: 'self' data: blob: internal-soap.wikia.com internal-soap.fandom.com internal-soap.wikia.org internal-soap.gamepedia.com www.fandom.com www.wikia.com www.wikia.org www.gamepedia.com; style-src 'self' data: blob: https://images.wikia.com https://static.wikia.nocookie.net https: 'self' data: blob: internal-soap.wikia.com internal-soap.fandom.com internal-soap.wikia.org internal-soap.gamepedia.com www.fandom.com www.wikia.com www.wikia.org www.gamepedia.com 'unsafe-inline'; img-src * data: blob; report-uri https://services.fandom.com/csp-logger/csp/ucp; worker-src 'self' blob;
last-modified	Sun, 13 Mar 2022 02:06:54 GMT
x-envoy-upstream-service-time	165
x-datacenter	SJC
x-cacheable	YES
accept-ranges	bytes

- ¿Cuál es la diferencia entre los códigos 200, 204, 304, 500?

Status Code 200: Es una respuesta aprobada estándar para solicitudes HTTP.

Status Code 204: El servidor procesó de forma exitosa la solicitud pero no retorna ningún contenido.

Status Code 304: Simplemente indica que el código o el recurso no ha sido modificado y por lo tanto no es necesario generar una copia.

Status Code 500: Ese apartado nos notifica que hay un problema con el servidor.

2. Entre a los sitios de Facebook analice la autenticación.

<https://www.facebook.com/Ninty64/photos/a.567021190154491/1752744198248845/>

Modificaciones:

- Se elimina la sección **/potos/** y como resultado se obtiene es que la página no existe.
- Se reemplaza el último número del enlace por un **6** y ocurre un error tipográfico por lo que no me lleva a nada.
- Por último, cambié la parte de **Ninty64** que es el nombre de la página por otra como: **WikiMyhero** y no hubo cambios.

3. Entre a Google Maps mire los parámetros de la url cámbielos y analice resultados cualesson los parámetros que pudo cambiar y cuales aparecen con las funcionalidades.

<https://www.google.com/maps/@15,-2.970703,3z?hl=es>

Esa es la URL inicial que nos aparece al entrar en la página de Google maps, entonces, comenzamos:

- Se modifíco la última sección **es** por **en** donde básicamente me retornó la misma página, pero en el idioma inglés.
- La ruta **@15,-2.970703,3** fue cambiada por los siguientes números: **@20-3.50,5** y obtuvimos una sección del mapa de Cartagena ,sin embargo, la ordenque mandé se cambió por **@10.387864,-75.5160121,15** por lo que asumo que se dirige a la ubicación donde se encuentra el dispositivo.

<https://www.google.com/maps/@10.387864,-75.5160121,15z?hl=en>

- Por último, se cambió el **15z** de esta última expresión por un **30z**, lo que ocasionó un incremento del mapa en un punto, por lo que se determina sencillamente que es la variable del aumento o de la disminución de la imagen.

Una de las funciones que tiene Google Maps es la de filtrar los resultados que aparecen en el mapa, por ejemplo, si se solicita que solo aparezcan farmacias:

<https://www.google.com/maps/search/Farmacias/@10.39314,-75.5093173,15z/data=!3m1!4b1>

- Como se puede evidenciar, entra en juego las variables **Farmacias** y **data=**.
- La ruta muestra farmacias cerca de la ubicación del dispositivo en uso.
- Se puede cambiar la palabra **Farmacias** por algún otro local que ofrece Google como **Restaurantes**.

4. **Entre a la siguiente dirección:**

<https://docs.postmanecho.com/?version=latest>

Use <https://postman-echo.com>

Realice:

- GET GET Request
- POST POST Raw Text
- POST POST Form Data
- PUT PUT Request
- PATCH PATCH Request
- DEL DELETE Request

Analice cada comando y que función tiene dentro de una página web.

GET GET request:

Es un comando utilizado para solicitar datos de un servidor.

GET <https://postman-echo.com/get?foo1=bar1&foo2=bar2> Send

Params Authorization Headers (6) Body Pre-request Script Tests Settings Cookies

Query Params

KEY	VALUE	DESCRIPTION
<input checked="" type="checkbox"/> foo1	bar1	
<input checked="" type="checkbox"/> foo2	bar2	
Key	Value	Description

Body Cookies (1) Headers (7) Test Results Status: 200 OK Time: 759 ms Size: 884 B Save Response

Pretty Raw Preview Visualize JSON

```
1 {
2   "args": {
3     "foo1": "bar1",
4     "foo2": "bar2"
5   },
6   "headers": {
7     "x-forwarded-proto": "https",
8     "x-forwarded-port": "443",
9     "host": "postman-echo.com",
10    "x-amzn-trace-id": "Root=1-61a6686d-6c2792f86707e1ba6dd0e95a",
11    "content-length": "9",
12    "user-agent": "PostmanRuntime/7.28.4",
13    "accept": "*/*",
14    "cache-control": "no-cache",
15    "postman-token": "e7e2d0df-e66d-4d66-aa76-2ee266fa8f6e",
16    "accept-encoding": "gzip, deflate, br",
17    "cookie": "sails.sig=s83A3Y1VQC6lmg5CKYwY140IS0t-Hjy8wwb.8bxk2FfcP39kCth6i1LLpod8ADn2FMM8oby2j5nCAQes5w"
18  },
19   "url": "https://postman-echo.com/get?foo1=bar1&foo2=bar2"
20 }
```

POST POST Raw Text

Este comando es utilizado para enviar información al servidor para crear o actualizar un recurso.

POST

https://postman-echo.com/post

Params

Authorization

Headers (7)

Body

Pre-request Script

Tests

Settings

Query Params

	KEY	VALUE	DESCRIPTION
<div><div></div><div></div></div>	a	b	c
	Key	Value	Description

Body

Cookies (1)

Headers (7)

Test Results

Status: 200 OK

Tim

Pretty

Raw

Preview

Visualize

JSON

1

2

3

4

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

"args": {},

"data": {},

"files": {},

"form": {},

"headers": {

"x-forwarded-proto": "https",

"x-forwarded-port": "443",

"host": "postman-echo.com",

"x-amzn-trace-id": "Root=1-61a65a3f-4660cb167f16e6a1206205cb",

"content-length": "0",

"user-agent": "PostmanRuntime/7.28.4",

"accept": "*/*",

"cache-control": "no-cache",

"postman-token": "e793cf74-33b8-4b7e-b815-5e66a44a931e",

"accept-encoding": "gzip, deflate, br",

"cookie": "sails.sid=s%3At34VaVLn-zINTiHYforizIxVJDGMxggL.kFGhJQZ9wjQx%2Fpbau4ctFtqiFgCdeiYvhyMRnvzaE3g"

},

"json": null,

"url": "https://postman-echo.com/post"

POST

https://postman-echo.com/post?a=b

Send

Params

Authorization

Headers (7)

Body

Pre-request Script

Tests

Settings

Query Params

	KEY	VALUE	DESCRIPTION
<div><div></div><div></div></div>	a	b	c
	Key	Value	Description

Body

Cookies (1)

Headers (7)

Test Results

Status: 200 OK

Time: 143 ms

Size: 890 B

Save Res

Pretty

Raw

Preview

Visualize

JSON

1

2

3

4

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

"args": {

"a": "b"

},

"data": {},

"files": {},

"form": {},

"headers": {

"x-forwarded-proto": "https",

"x-forwarded-port": "443",

"host": "postman-echo.com",

"x-amzn-trace-id": "Root=1-61a65a4e-004fa71f933602863d9ec8fe",

"content-length": "0",

"user-agent": "PostmanRuntime/7.28.4",

"accept": "*/*",

"cache-control": "no-cache",

"postman-token": "a7513594-d727-433d-862b-7b55883e640e",

"accept-encoding": "gzip, deflate, br",

"cookie": "sails.sid=s%3AQ9VnFpDR4Pzovic09g4gsaKF81y9TFVy.Csjh9j9Uc1JVPmIRbdfbxexuQNF3mL8SHutwJlMn1uQ"

},

"json": null,

POST form data

Esta función está diseñada para poder enlistar un # de datos determinados y enviarlos al servidor.

The screenshot shows a Postman interface for a POST request to `https://postman-echo.com/post`. The request is configured with form data. The body tab is active, showing the raw JSON response from the server.

Request Configuration:

- Method: POST
- URL: `https://postman-echo.com/post`
- Body Type: form-data

Form Data:

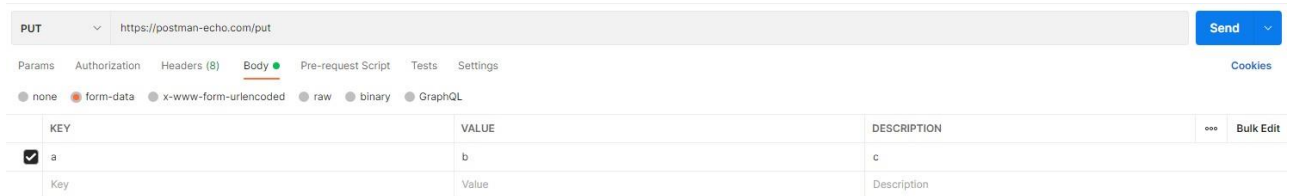
KEY	VALUE	DESCRIPTION
a	b	c
Key	Value	Description

Response (JSON):

```
1 {
2   "args": {},
3   "data": {},
4   "files": {},
5   "form": {
6     "a": "b"
7   },
8   "headers": {
9     "x-forwarded-proto": "https",
10    "x-forwarded-port": "443",
11    "host": "postman-echo.com",
12    "x-amzn-trace-id": "Root=1-61a65b18-47324af26ebf6d47765ea8ce",
13    "content-length": "133",
14    "user-agent": "PostmanRuntime/7.28.4",
15    "accept": "*/*",
16    "cache-control": "no-cache",
17    "postman-token": "8943f381-3756-4bbf-a1f4-b38c28cfb239",
18    "accept-encoding": "gzip, deflate, br",
19    "connection": "keep-alive"
20  }
21 }
```


PUT PUT Request:

Este comando como el POST es utilizado para enviar datos al servidor con el fin de crear o actualizar datos, lo que diferencia este al POST es que si uno envía el mismo PUT Request varias veces esto solo producirá el mismo resultado, en cambio, el POST Request crearía diferentes recursos múltiples veces.



The screenshot shows the Postman interface for a PUT request to `https://postman-echo.com/put`. The 'Body' tab is selected, and the 'form-data' type is chosen. A table with two columns, 'KEY' and 'VALUE', contains one entry: 'a' with value 'b'. The 'Send' button is in the top right corner.

KEY	VALUE	DESCRIPTION
a	b	c
Key	Value	Description



The screenshot shows the 'Body' tab of the response for the PUT request. The response is in JSON format, showing the request details as received by the server. The status is 200 OK, and the response size is 971 B.

```
1 {
2   "args": {},
3   "data": {},
4   "files": {},
5   "form": {
6     "a": "b"
7   },
8   "headers": {
9     "x-forwarded-proto": "https",
10    "x-forwarded-port": "443",
11    "host": "postman-echo.com",
12    "x-amzn-trace-id": "Root=1-61a65d5a-6857929d6fb05242274786ca",
13    "content-length": "133",
14    "user-agent": "PostmanRuntime/7.28.4",
15    "accept": "*/*",
16    "cache-control": "no-cache",
17    "postman-token": "72983ad1-2996-4783-8411-6c3d5a769f50",
18    "accept-encoding": "gzip, deflate, br",
19    "cookie": "sails.sid=s%3Avf2Ch1M116pjr1duzXSE-IQYXIBebJ1.IRSMjtukyjo0d2E10%2Fu4737PwHKAm1InaoW1cjrjLzq",
20    "content-type": "multipart/form-data; boundary=----WebKitFormBoundary1SCsTwhM11293zPw"
21  }
```

PATCH PATCH Request

Su función es similar al PUT debido a que se encarga de enviar datos y/o cambiarlos en un servidor, pero, a diferencia del PUT, PATCH no es idempotente.



The screenshot shows the Postman interface for a PATCH request to `https://postman-echo.com/patch`. The 'Body' tab is selected, and the 'form-data' type is chosen. A table with two columns, 'KEY' and 'VALUE', contains one entry: 'a' with value 'b'. The 'Send' button is in the top right corner.

KEY	VALUE	DESCRIPTION
a	b	c
Key	Value	Description



The screenshot shows the 'Body' tab of the response for the PATCH request. The response is in JSON format, showing the request details as received by the server. The status is 200 OK, and the response size is 977 B.

```
1 {
2   "args": {},
3   "data": {},
4   "files": {},
5   "form": {
6     "a": "b"
7   },
8   "headers": {
9     "x-forwarded-proto": "https",
10    "x-forwarded-port": "443",
11    "host": "postman-echo.com",
12    "x-amzn-trace-id": "Root=1-61a88ef9-8c4bd3168d56396a33496f96",
13    "content-length": "133",
14    "user-agent": "PostmanRuntime/7.28.4",
15    "accept": "*/*",
16    "cache-control": "no-cache",
17    "postman-token": "1d86a726-ecb2-4943-9969-79185c1afa93",
18    "accept-encoding": "gzip, deflate, br",
19    "cookie": "sails.sid=s%3A8auwqNMGbaHbBNPbC963HPgXV8gkS1.8d9CCNuNaTa4XnS1efctagW2BC2n3wvvnTaZ1282F5it04k",
20    "content-type": "multipart/form-data; boundary=----WebKitFormBoundary6fduZvY66Pw9Ww1E"
21  }
```

DEL DELETE Request

Básicamente se encarga en eliminar una fuente determinada.

The screenshot displays a Postman interface for a DELETE request to `https://postman-echo.com/delete`. The request is configured with the following details:

- Method:** DELETE
- URL:** `https://postman-echo.com/delete`
- Body:** form-data
- Headers:** 6 headers are listed in the response body.

The response body is shown in the "Body" tab, displaying the following JSON structure:

```
1 {
2   "args": {},
3   "data": {},
4   "files": {},
5   "form": {},
6   "headers": {
7     "x-forwarded-proto": "https",
8     "x-forwarded-port": "443",
9     "host": "postman-echo.com",
10    "x-amzn-trace-id": "Root=1-61a88fe9-715e4ed71b36d7693e8dd7c9",
11    "content-length": "0",
12    "user-agent": "PostmanRuntime/7.28.4",
13    "accept": "*/*",
14    "cache-control": "no-cache",
15    "postman-token": "16f96b8a-756d-471a-a1fe-ca870c2f01e2",
16    "accept-encoding": "gzip, deflate, br",
17    "cookie": "sails.sid=s%3AYVRnMhOaVreA7I9oa4U6I1a5DLmXbhg.jwyImKuFAf10W6OiwAw13%2F1V4ESW0RNFm10lKhbroqo"
18  },
19   "json": null,
20   "url": "https://postman-echo.com/delete"
21 }
```

The status bar at the bottom indicates a successful response with a status of 200 OK, a time of 141 ms, and a size of 883 B. The "Save Response" button is visible.

5. De postman-echo realice un get de repuesta json, analice la respuesta. Ahora compárelo contra el siguiente: linkjsongis

¿Qué diferencias en los parámetros encuentra y que diferencia en la respuesta?

Se puede evidenciar que se genera respuestas con html y json con respecto a ciertas funciones, botones, o partes que posee la página postman echo.

```
<!DOCTYPE html>
<html>
  <head>...</head>
  <body style="overflow: unset;" class>
    <link rel="stylesheet" type="text/css" href="https://st-ar.cdn.postman.com/orm-d3d2716...css">
    <link rel="stylesheet" as="style" href="https://st-ar.cdn.postman.com/legacy-f8bef61...css" id="requester-styles" type="text/css">
    <style id="theme">...</style>
    <div class="pm-loader is-hidden">...</div>
    <div id="postman_preload_cache" data-preload-cache="{ }"></div>
    <script nonce>...</script>
    <div class="app-root" id="app-root" data-theme="light">
      <div class="app-requester" tabindex="-1" style="min-width: 1130px;">
        <div class="app-requester" style="min-width: 1130px;">
          <div id="dropdown-root" style="width: 0px; height: 0px;"></div>
          <div id="auto-suggest-root" style="width: 0px; height: 0px;"></div>
          <div id="right-overlay-root"></div>
          <div id="overlay-root"></div>
          <div class="requester-header" tabindex="-1">...</div>
          <div class="pane-group vertical">
            <div class="pane-container">
              <div class="pane resizableX resizableY requester-contents-pa"
                "height: 897px; width: 1401px;">
                <div class="pane-content">
```

6. Metodos de Autenticación realice en postman-echo GET Basic AuthAnalice los parametros.

Este es un método de autorización donde se exige el usuario y la contraseña, devolviendo el estado 200 OK en caso de estar autorizados y estado 401 en caso de que no.

The screenshot shows the Postman interface for a GET request to `https://postman-echo.com/basic-auth`. The **Authorization** tab is selected, showing **Basic Auth** type. The **Username** is `postman` and the **Password** is masked with dots. A warning message states: "Heads up! These parameters hold sensitive data. To keep this data secure while working in a collaborative environment, we recommend using variables. Learn more about variables". The **Body** tab is selected, showing the response in **Pretty** format: `{\"authenticated\": true}`. The status bar at the bottom indicates **Status: 200 OK**, **Time: 124 ms**, and **Size: 358 B**.

The screenshot shows the Postman interface for a GET request to `https://postman-echo.com/basic-auth`. The **Authorization** tab is selected, showing **Basic Auth** type. The **Username** is `pepito` and the **Password** is masked with dots. A warning message states: "Heads up! These parameters hold sensitive data. To keep this data secure while working in a collaborative environment, we recommend using variables. Learn more about variables". The **Body** tab is selected, showing the response in **Text** format: `Unauthorized`. The status bar at the bottom indicates **Status: 401 Unauthorized**, **Time: 97 ms**, and **Size: 293 B**.