

Instituto Politécnico Nacional



Integrantes:

Álvarez Hernández Eduardo

Daniel Rubio Ramón

Trejo Rivera Oscar Gerardo

Materia:

Aplicaciones para comunicaciones de Red

“Reporte Practica 4 Apache”

Grupo: 3CM8

Fecha de entrega: 6 de Junio de 2019

Introducción

Si bien hay una gran complejidad que apunta a cómo funciona un servidor web, el trabajo básico de todos los servidores web es aceptar solicitudes de clientes (por ejemplo, el navegador web de un visitante) y luego enviar la respuesta a esa solicitud (por ejemplo, los componentes de la página que el visitante quiere ver).

Desarrollo

En esta práctica se hicieron peticiones como clientes a un servidor, que actúa como un apache, nos brinda un servicio con el método POST y GET.

Como primer lugar veamos que son el método GET:

Cuando un usuario rellena un formulario en una página web los datos hay que enviarlos de alguna manera. Vamos a considerar las dos formas de envío de datos posibles: usando el método POST

Por

ejemplo: `<form action="http://www.aprenderaprogramar.com/prog/newuser" method="get">`

En el ejemplo anterior la acción que se ejecutará cuando el usuario pulse el botón “Enviar” (submit) será el envío de los datos a la url especificada usando el método get.

Veamos el aspecto de un formulario cualquiera para hacernos una idea general.



Formulario de comunicación

Puedes contactar con nosotros a través de este formulario o usando el correo electrónico contacto@aprenderaprogramar.com

Obligatorio*

Nombre: * ⓘ

Apellidos: ⓘ

Correo electrónico: ⓘ

País: * ⓘ

Mensaje: * ⓘ

Este formulario consta de varios campos que al usuario se le solicitan como Nombre, Apellidos, Correo electrónico, País y Mensaje. Posiblemente los nombres de los campos

en el código HTML sean del tipo nombre_user, apellidos_user, email_user, pais_user y msg.

La diferencia entre los métodos get y post radica en la forma de enviar los datos a la página cuando se pulsa el botón "Enviar". Mientras que el método GET envía los datos usando la URL, el método POST los envía de forma que no podemos verlos (en un segundo plano u "ocultos" al usuario).

Un resultado usando el método GET, a modo de ejemplo, podría ser el siguiente:

<http://www.aprenderaprogramar.com/newuser.php?nombre=Pepe&apellido=Flores&email=h52turam%40uco.es&sexo=Mujer>

El símbolo ? indica dónde empiezan los parámetros que se reciben desde el formulario que ha enviado los datos a la página.

Ahora veamos el método POST:

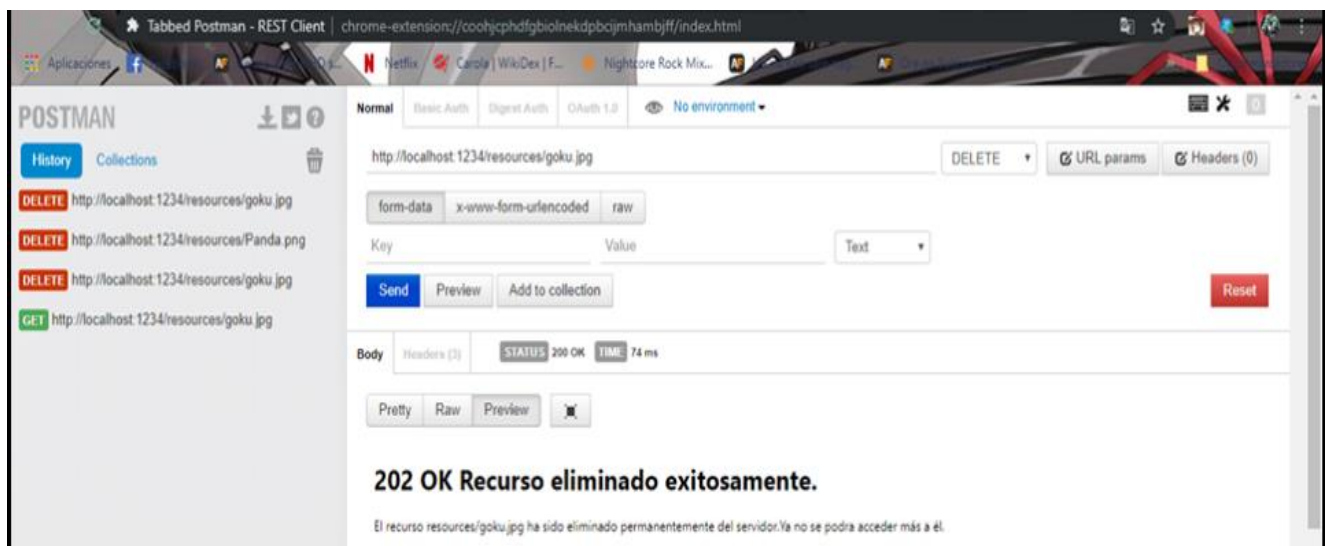
Hemos visto el resultado de un envío por el método GET. En el caso de un envío de datos usando el método POST, aunque estos datos también serán enviados (de una forma que podemos denominar "oculta"), no los podremos ver en la URL. Para poder recuperar los valores de los campos en el caso de un envío con el método POST necesitaríamos otras herramientas (por ejemplo, valernos del lenguaje PHP para recuperar el valor de esos campos).

El resultado final con ambos métodos podemos decir que es el mismo: la información se transmite de un lado a otro. La diferencia radica en que con el método GET podemos ver directamente los parámetros pasados ya que están dentro de la URL mientras que con el método POST los parámetros quedan ocultos y para rescatarlos hay que usar otras herramientas.

Un ejemplo de uso del método post sería este:

`<form action="http://www.aprenderaprogramar.com/prog/newuser" method="post">`

Al hacer peticiones, necesitamos de un cliente, para poder ver mas a fondo de como funciona esta practica se uso POSTMAN que es una extensión del navegador Google Chrome, que permite el envío de peticiones HTTP REST sin necesidad de desarrollar un cliente.



Y como primer lugar se hicieron dos formularios, uno que funciona con el método POST y otro que funciona con el método GET

localhost:1234/index.html

Recursos disponibles

Formulario GET

Nombre: Eduardo
Dirección: slkndsajU40932849
Teléfono: u509'932'4
Comentarios: sksnf ewjrev

Enviar datos Borrar datos

Formulario POST

Nombre:
Dirección:
Teléfono:
Comentarios:

A continuación, se muestran los parámetros obtenidos por el método GET:

localhost:1234/get?Nombre=Eduardo&Direccion=slkndsajU4093284%279&Telefono=u509%27932%274&Comentarios=sksnf+ewjrev

Parametros obtenidos por medio de GET

Parametro	Valor
Nombre	Eduardo
Direccion	slkndsajU4093284%279
Telefono	u509%27932%274
Comentarios	sksnf+ewjrev

Y por el método POST:

localhost:1234/post

Parametros obtenidos por medio de POST

Parametro	Valor
Nombre	Eduardo%7C
Direccion	shdoisajdimk
Telefono	2840384%279213%27
Comentarios	lmf+opereov+jieijioe+ceojm

Por ultimo se muestran las peticiones de los clientes, estos se pueden observar cuando se corre el servidor, la siguiente captura muestra como cada cliente solicita consultas por método POST o GET:

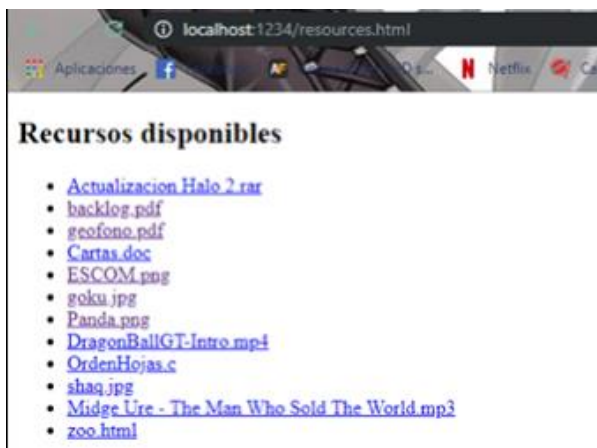
```
C:\Windows\System32\cmd.exe - java ServidorWeb
Espanando a Cliente....
-----> Cliente Conectado desde: /0:0:0:0:0:0:1
Por el puerto: 52959
Datos: GET /index.html HTTP/1.1

Respuesta GET:
HTTP/1.1 200 OK
Date: Tue Jun 04 19:42:32 CDT 2019
Server: EnrikeAbi Server/1.0
Content-Type: text/html
Content-Length: 1182

-----> Cliente Conectado desde: /0:0:0:0:0:0:1
Por el puerto: 52958
Datos: GET /favicon.ico HTTP/1.1

Respuesta GET:
HTTP/1.1 404 Not Found
Date: Tue Jun 04 19:42:33 CDT 2019
Server: EnrikeAbi Server/1.0
Content-Type: text/html
Content-Length: 215
```

Así mismo el servidor proporciona recursos disponibles como lo son imágenes, música, etc, que el usuario puede visualizar o acceder a ellos.



Y si el servidor borra uno de estos archivos y un cliente quiere acceder a él se muestra:

