

¿Cómo funciona un servidor web explicado desde PowerShell?

El protocolo utilizado para la transmisión de archivos es HTTP o HTTPS (ofrece un nivel extra de seguridad gracias al cifrado). Este protocolo se basa a su vez en los protocolos de red IP y TCP. Así, un servidor web puede mostrar el contenido de un sitio web de forma simultánea a varios clientes que entiendan el protocolo HTTP.

HTTP define un conjunto de métodos de petición para indicar la acción que se desea realizar para un recurso determinado. Aunque estos también pueden ser sustantivos, estos métodos de solicitud a veces son llamados HTTP verbs.

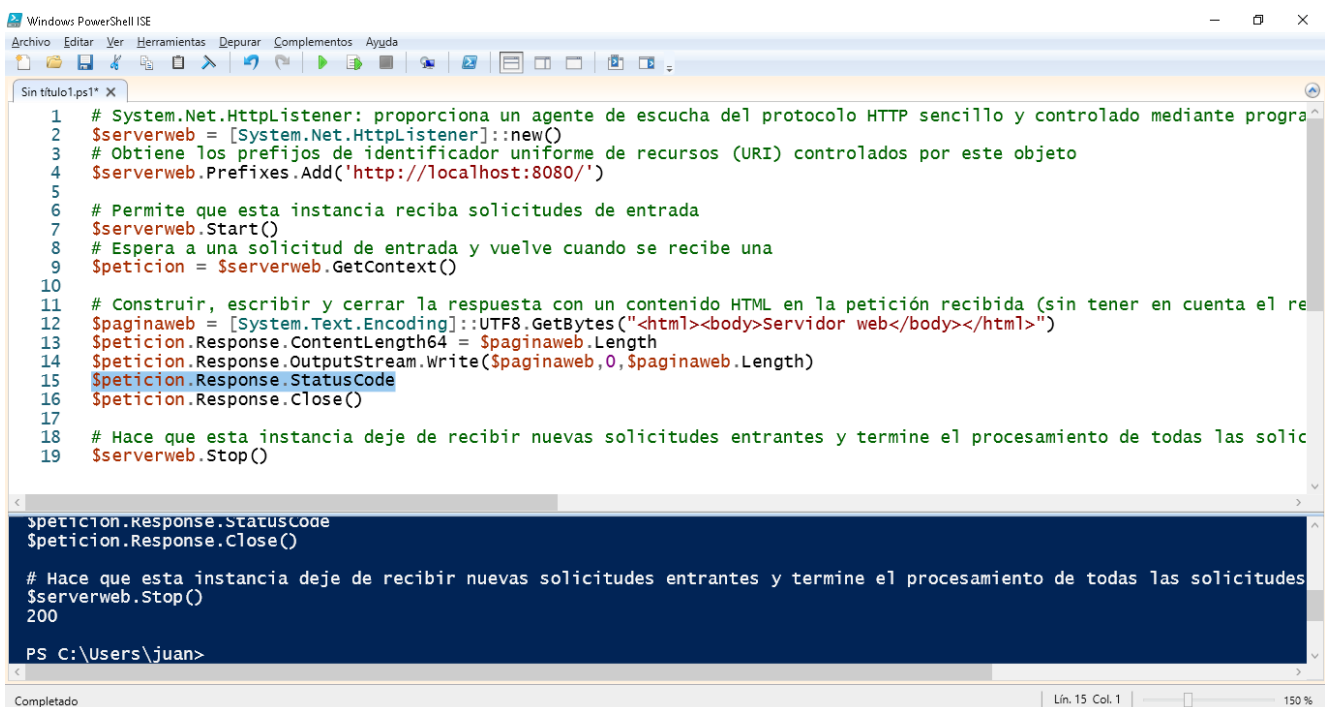
- GET. El método GET solicita una representación de un recurso específico. Las peticiones que usan el método GET sólo deben recuperar datos.
- HEAD. El método HEAD pide una respuesta idéntica a la de una petición GET, pero sin el cuerpo de la respuesta.
- POST. El método POST se utiliza para enviar una entidad a un recurso en específico, causando a menudo un cambio en el estado o efectos secundarios en el servidor.
- PUT. El modo PUT reemplaza todas las representaciones actuales del recurso de destino con la carga útil de la petición.
- DELETE. El método DELETE borra un recurso en específico.
- CONNECT. El método CONNECT establece un túnel hacia el servidor identificado por el recurso.
- OPTIONS. El método OPTIONS es utilizado para describir las opciones de comunicación para el recurso de destino.
- TRACE. El método TRACE realiza una prueba de bucle de retorno de mensaje a lo largo de la ruta al recurso de destino.

- PATCH. El método PATCH es utilizado para aplicar modificaciones parciales a un recurso.

Los mensajes HTTP, son los medios por los cuales se intercambian datos entre servidores y clientes. Hay dos tipos de mensajes: peticiones, enviadas por el cliente al servidor, para pedir el inicio de una acción; y respuestas, que son la respuesta del servidor.

Los mensajes HTTP están compuestos de texto, codificado en ASCII, y pueden comprender múltiples líneas. En HTTP/1.1, y versiones previas del protocolo, estos mensajes eran enviados de forma abierta a través de la conexión. En HTTP/2.0 los mensajes, que anteriormente eran legibles directamente, se conforman mediante tramas binarias codificadas para aumentar la optimización y rendimiento de la transmisión.

[crayon-6a9d6f527c872517858946/]



```
Windows PowerShell ISE
Archivo Editar Ver Herramientas Depurar Complementos Ayuda

Sin título1.ps1* X
1 # System.Net.HttpListener: proporciona un agente de escucha del protocolo HTTP sencillo y controlado mediante programa
2 $serverweb = [System.Net.HttpListener]::new()
3 # Obtiene los prefijos de identificador uniforme de recursos (URI) controlados por este objeto
4 $serverweb.Prefixes.Add('http://localhost:8080/')
5
6 # Permite que esta instancia reciba solicitudes de entrada
7 $serverweb.Start()
8 # Espera a una solicitud de entrada y vuelve cuando se recibe una
9 $peticion = $serverweb.GetContext()
10
11 # Construir, escribir y cerrar la respuesta con un contenido HTML en la petición recibida (sin tener en cuenta el re
12 $paginaweb = [System.Text.Encoding]::UTF8.GetBytes("<html><body>Servidor web</body></html>")
13 $peticion.Response.ContentLength64 = $paginaweb.Length
14 $peticion.Response.OutputStream.Write($paginaweb,0,$paginaweb.Length)
15 $peticion.Response.StatusCode
16 $peticion.Response.Close()
17
18 # Hace que esta instancia deje de recibir nuevas solicitudes entrantes y termine el procesamiento de todas las solici
19 $serverweb.Stop()

$peticion.Response.StatusCode
$peticion.Response.Close()

# Hace que esta instancia deje de recibir nuevas solicitudes entrantes y termine el procesamiento de todas las solicitudes
$serverweb.Stop()
200

PS C:\Users\juan>
```



Servidor web