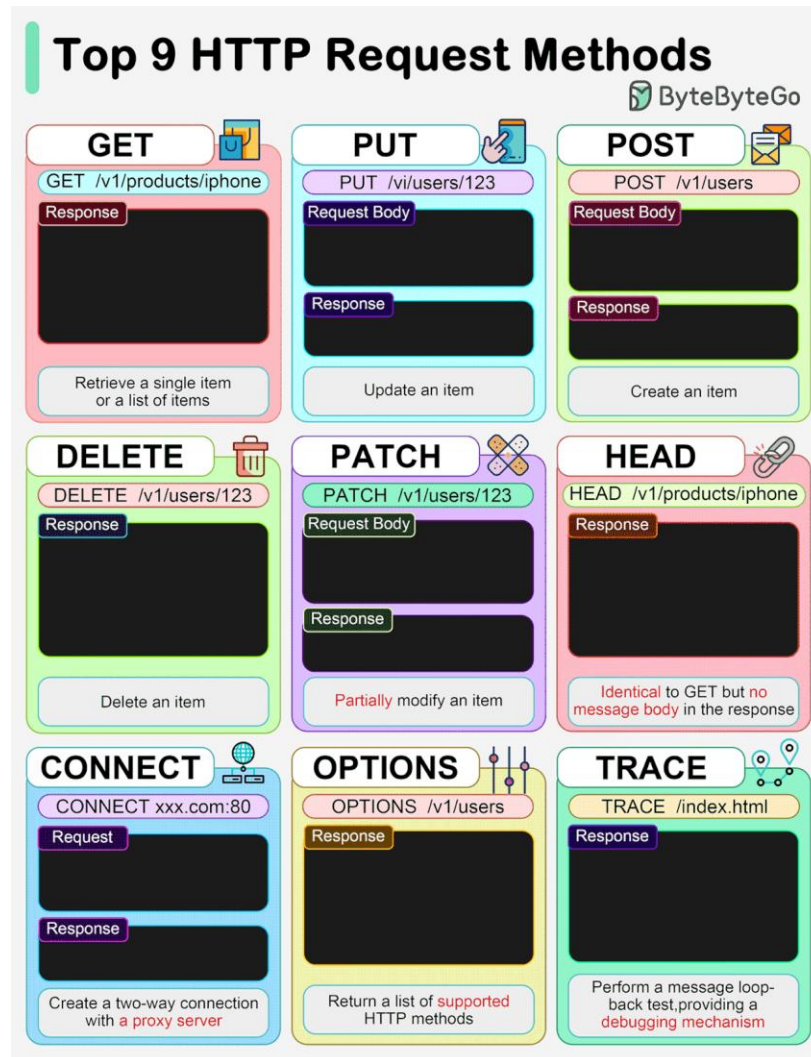


Actividad : HTTP

Guía de Introducción al Protocolo HTTP



Objetivo:

- Comprender los conceptos básicos del Protocolo de Transferencia de Hipertexto (HTTP).
- Familiarizarse con los métodos HTTP más comunes: GET, POST, PUT y DELETE.
- Aplicar los conocimientos adquiridos a través de ejercicios prácticos.

¿Qué es HTTP? (deben complementar la información presentada con consultas adicionales)

- HTTP es el Protocolo de Transferencia de Hipertexto, utilizado para la comunicación entre clientes (navegadores web) y servidores.
- Define cómo se envían y reciben solicitudes y respuestas entre el cliente y el servidor.

- Basado en el modelo cliente-servidor.
- **Stateless:** Cada solicitud se procesa de forma independiente, sin tener en cuenta las solicitudes anteriores.
- **Sin conexión:** Cada solicitud y respuesta son independientes y no mantienen una conexión persistente.

Métodos HTTP

Los métodos HTTP especifican la acción que se debe realizar en el recurso identificado por la URL. Los métodos más comunes son:

1. **GET:** Solicita datos del servidor. Es utilizado para recuperar información.
2. **POST:** Envía datos al servidor. Es utilizado para enviar datos al servidor para su procesamiento.
3. **PUT:** Actualiza un recurso en el servidor. Es utilizado para actualizar información existente en el servidor.
4. **DELETE:** Elimina un recurso en el servidor. Es utilizado para eliminar información del servidor.

Ejercicios : (Usar ejemplos adjuntos como referencia)

Ejercicio 1: Simular solicitudes GET y POST

1. Utiliza un navegador web para enviar una solicitud GET a una página web de tu elección.
2. Observa la respuesta del servidor.
3. Utiliza una herramienta como Wireshark para inspeccionar las solicitudes.

Ejercicio 2: Implementar solicitudes GET y POST en un formulario HTML

1. Crea un archivo HTML con un formulario que envíe una solicitud GET a una URL específica.
2. Crea otro formulario que envíe una solicitud POST a la misma URL.
3. Utiliza una herramienta como Wireshark para inspeccionar las solicitudes y respuestas HTTP generadas por los formularios.

Ejercicio 3: Explorar otros métodos HTTP

1. Investiga sobre los métodos PUT y DELETE.
2. Intenta enviar solicitudes PUT y DELETE utilizando herramientas como Postman o cURL.

Recursos Adicionales

- [Mozilla Developer Network \(MDN\) - HTTP](#)
- [Postman](#)
- [cURL](#)

Ejemplos:

Ejemplo 1: html+css:

En el ejemplo proporcionado el servidor escucha en el puerto 8080 y responde a las solicitudes de los clientes con una página web compuesta por un archivo HTML (`index.html`), un archivo CSS (`styles.css`) y dos imágenes (`image1.jpg` y `image2.jpg`). Asegúrate de crear una carpeta llamada "**content**" en el mismo directorio que el archivo `Server.java`, y coloca los archivos HTML, CSS e imágenes dentro de esta carpeta.

Este ejemplo utiliza la codificación base64 para incrustar el contenido de las imágenes directamente en la respuesta HTTP. **Tener en cuenta que este enfoque puede no ser adecuado para imágenes grandes debido a la sobrecarga de codificación base64.** En la práctica, sería más eficiente enviar las imágenes como archivos adjuntos y referenciarlas en el HTML usando etiquetas `` estándar. **Investigar que modificaciones se deben hacer para implementar este enfoque.**

Ejecución del Proyecto:

Para ejecutar el proyecto, sigue estos pasos inicialmente en localhost:

1. Coloca los archivos `index.html` y `styles.css` en el directorio **content**.
2. Compilar y ejecutar el código en java proporcionado `Server.java`
3. Abrir un navegador web y e ingresa <http://localhost:8080> . Deberás cambiar la **ip** en caso de desplegar el proyecto en una maquina virtual o un servidor remoto.
4. Si deseas realizar cambios en la página, puedes editar los archivos HTML y CSS según tus necesidades.

Ejemplo 2: Formulario HTML para una solicitud GET

En este ejemplo se proporciona la implementación de un servidor en Java capaz de manejar solicitudes típicas como POST y GET. También se les presenta un código HTML para ilustrar el uso de estos métodos. Este HTML contiene dos formularios, uno para enviar una solicitud GET y otro para enviar una solicitud POST al servidor Java que hemos creado. Al hacer clic en los botones "Enviar GET" o "Enviar POST", se enviará un mensaje al servidor, y el servidor responderá con una página que muestra los datos recibidos. Deben personalizar y complementar los ejemplos.

Se ejecuta el `HTTPServer` y luego en el navegador abres <http://localhost:8080>