

URL <https://g.co/gemini/share/f832da63a93d>

Código para usar fetch con un requerimiento GET en una aplicación web:

JavaScript

```
async function obtenerDatos() {
  const url = "https://www.ejemplo.com/"; // Reemplaza con la URL real
  de la API o recurso
  const respuesta = await fetch(url);

  if (!respuesta.ok) {
    console.error("Error:", respuesta.status);
    return;
  }

  const datos = await respuesta.json();
  console.log(datos); // Procesar o mostrar los datos obtenidos
}

obtenerDatos();
```

Explicación del código:

1. **Función obtenerDatos:** Esta función define la lógica para realizar la solicitud GET y procesar la respuesta.
2. **const url = "https://www.ejemplo.com/";**: Se define la URL del recurso o API al que se desea realizar la solicitud GET. Reemplaza `https://www.ejemplo.com/` con la URL real.
3. **const respuesta = await fetch(url);**: Se utiliza la función `fetch` para realizar la solicitud GET a la URL especificada. La respuesta se almacena en la variable `respuesta`.
4. **if (!respuesta.ok) { ... }**: Se verifica si la solicitud fue exitosa. Si no lo fue (`respuesta.ok` es `false`), se muestra un mensaje de error en la consola y se termina la función.
5. **const datos = await respuesta.json();**: Si la solicitud fue exitosa, se extrae el cuerpo de la respuesta en formato JSON y se almacena en la variable `datos`.
6. **console.log(datos);**: Se muestra la información obtenida en la consola. Puedes reemplazar esto con la lógica que necesites para procesar y mostrar los datos en tu aplicación web.
7. **obtenerDatos();**: Se llama a la función `obtenerDatos` para iniciar el proceso de solicitud y procesamiento de datos.

Consideraciones adicionales:

- Asegúrate de incluir la biblioteca `fetch` en tu proyecto. En la mayoría de los navegadores modernos está disponible de forma nativa, pero puedes verificarlo e incluirla si es necesario.
- Puedes modificar la función `obtenerDatos` para agregar más lógica, como manejar errores

específicos, procesar los datos de forma diferente o mostrarlos en la interfaz de usuario de tu aplicación web.

- Es importante recordar que las solicitudes GET se utilizan para recuperar datos, no para modificarlos o crear nuevos datos. Para eso, se utilizan otros métodos HTTP como POST o PUT.

Recursos adicionales:

- [MDN Web Docs: fetch](#)
- [Ejemplos de uso de fetch](#)