

Bing

¡Por supuesto! Permíteme proporcionarte un ejemplo más detallado y realista de cómo implementar un **Singleton** en **JavaScript**. Utilizaré la versión basada en módulos para mayor claridad.

Supongamos que estamos creando una aplicación web y necesitamos una única instancia para manejar la configuración global de la aplicación. Aquí está el código:

```
// ConfigModule.js

const ConfigModule = (function () {
  // Variables privadas
  let instance;
  const configData = {
    apiUrl: 'https://api.example.com',
    apiKey: 'your-api-key',
    debugMode: false,
  };

  // Inicialización (solo se ejecuta una vez)
  function init() {
    // Lógica de inicialización aquí (por ejemplo, cargar configuraciones desde un archivo)
    return {
      getConfig: function () {
        return configData;
      },
      setDebugMode: function (value) {
        configData.debugMode = value;
      },
    };
  };

  // Método público para obtener la instancia
  return {
    getInstance: function () {
      if (!instance) {
        instance = init();
      }
      return instance;
    },
  };
})();

// Uso del Singleton
const configInstance = ConfigModule.getInstance();

// Obtener la configuración
const appConfig = configInstance.getConfig();
console.log('API URL:', appConfig.apiUrl);
console.log('API Key:', appConfig.apiKey);

// Cambiar al modo de depuración
configInstance.setDebugMode(true);
console.log('Debug Mode:', appConfig.debugMode);
```

En este ejemplo:

1. Creamos un módulo llamado `ConfigModule` que contiene la configuración global de nuestra aplicación.
2. La función `init` inicializa la configuración y proporciona métodos para acceder y modificar los valores.
3. El método `getInstance` garantiza que solo haya una instancia activa de `ConfigModule`.

4. Utilizamos `configInstance` para acceder a la configuración en cualquier parte de nuestra aplicación.

Recuerda adaptar este ejemplo a tus necesidades específicas. ¡Espero que te ayude a comprender mejor cómo implementar un Singleton en JavaScript! 😊.