

jsvelox — Guia Completa

Todo lo que necesita saber sobre jsvelox
v2.2.0 — Nombre en clave: Quagga

github.com/yunus6141/jsvelox

// 01

Que es jsvelox?

jsvelox es una biblioteca JavaScript ligera que le permite integrar proveedores de IA - Claude, OpenAI y Gemini - en sus proyectos con solo unas pocas lineas de codigo. A diferencia de las pesadas plataformas de agentes de IA que consumen recursos incluso para funciones no utilizadas, jsvelox sigue un principio simple: use solo lo que necesita.

Disenada para desarrolladores que quieren integracion de IA sin la sobrecarga de constructores de pipelines visuales o configuraciones complejas de servidor.

// 02

Que problemas resuelve jsvelox?

Demasiado complejo	Las plataformas de agentes de IA requieren interfaces visuales y curvas de aprendizaje empinadas.	✓ jsvelox usa objetos JavaScript simples - sin arrastrar y soltar.
Demasiado pesado	Plataformas como n8n consumen RAM para cada nodo, incluso los no utilizados.	✓ jsvelox solo usa recursos para lo que realmente llama.
Demasiados archivos	Integrar IA generalmente significa gestionar multiples clientes API.	✓ Una biblioteca, tres proveedores de IA, API unificada.
Difícil de personalizar	Entrenar una IA con sus propios datos requiere configuraciones complejas.	✓ systemPrompt y context permiten entrenar la IA en el mismo objeto de configuracion.

Estructura del codigo

jsvelox se basa en una sola clase: VeloxAI. Como funciona:

Configuracion requerida

```
import VeloxAI from 'jsvelox'

const ai = new VeloxAI({
  provider: 'gemini', // Requerido: claude | openai | gemini
  apiKey: 'su-clave', // Requerido: su clave API
  memoryLimit: 10,    // Opcional: max mensajes a recordar
  timeout: 15000      // Opcional: ms antes del timeout
})

const respuesta = await ai.send('Hola!')
console.log(respuesta)
```

Configuracion de entrenamiento de IA

```
const ai = new VeloxAI({
  provider: 'gemini',
  apiKey: 'su-clave',
  systemPrompt: 'Eres un asistente de tienda. Responde en espanol.',
  context: [
    { nombre: 'Producto A', precio: '29 EUR', beneficio: 'Aumenta energia' },
    { nombre: 'Producto B', precio: '49 EUR', beneficio: 'Mejora el sueno' }
  ]
})
```

Metodos disponibles

Method	Description
send(mensaje)	Enviar un mensaje y recibir una respuesta
sendWithRetry(msg, n)	Enviar con reintento automatico en caso de fallo (predeterminado: 3 veces)
onSend(btn, input, res)	Vincular boton/entrada/respuesta en una linea
getMemory()	Obtener historial de conversacion completo
clearMemory()	Borrar historial de conversacion

Errores comunes y soluciones

ERROR	Clave API requerida!
Cause:	VeloxAI creado sin apiKey.
Fix:	Siempre proporcione apiKey: <pre>new VeloxAI({ provider: 'gemini', apiKey: 'clave' })</pre>
ERROR	Proveedor requerido!
Cause:	No se especifico el proveedor de IA.
Fix:	Agregar provider: <pre>new VeloxAI({ provider: 'gemini', apiKey: 'clave' })</pre>
ERROR	Proveedor invalido!
Cause:	Nombre de proveedor no compatible. Ejemplo: provider: 'gpt'
Fix:	Solo use: claude openai gemini
ERROR	El mensaje no puede estar vacio!
Cause:	ai.send() llamado con cadena vacia.
Fix:	Verificar mensaje antes de enviar: <pre>if (msg.trim()) { await ai.send(msg) }</pre>
ADVERTE NCIA	Tiempo de espera agotado!
Cause:	La IA no respondio dentro del limite de tiempo.
Fix:	Aumentar timeout o usar sendWithRetry.
ERROR	Clave API invalida
Cause:	Su clave API es incorrecta o ha expirado.
Fix:	Obtener clave valida de console.anthropic.com, platform.openai.com o aistudio.google.com

ADVERTE NCIA	Cuota excedida
Cause:	Ha usado toda su cuota de API gratuita del día.
Fix:	Espere el reinicio de cuota o actualice su plan.
ERROR	Elementos no encontrados
Cause:	Los selectores CSS de onSend() no coinciden.
Fix:	Verificar selectores: ai.onSend('#btn', '#entrada', '#salida')
ADVERTE NCIA	Error CORS (navegador)
Cause:	API llamada directamente desde el navegador.
Fix:	Use jsvelox en backend Node.js, no directamente en el navegador.

// 06

Manejo de errores con try-catch

```
try {
  const respuesta = await ai.send('Hola!')
  console.log(respuesta)
} catch (error) {
  if (error.message.includes('Timeout')) {
    console.log('IA demasiado lenta, intentelo de nuevo')
  } else {
    console.log('Error:', error.message)
  }
}
```

// 07

Cuando funciona mejor jsvelox?

- ✓ Aplicaciones backend Node.js
- ✓ Servidores Express.js o Fastify
- ✓ Rutas API de Next.js
- ✓ Herramientas CLI y scripts
- ✓ Cualquier proyecto JavaScript/TypeScript en Node.js 18+

Cuando NO funciona jsvelox?

- ✗ Uso directo en navegador (restriccion CORS) - disponible en v3.0
- ✗ Entornos Deno o Bun (no probado)
- ✗ Versiones de Node.js inferiores a 18

// 08

Uso con TypeScript

```
import VeloxAI, { VeloxConfig } from 'jsvelox'

const config: VeloxConfig = {
  provider: 'gemini',
  apiKey: process.env.GEMINI_KEY!,
  memoryLimit: 20
}

const ai = new VeloxAI(config)
const respuesta: string = await ai.send('Hola!')
```

