

jsvelox — Tam Rehber

jsvelox hakkında bilmeniz gereken her sey
v2.2.0 — Kod Adi: Quagga

github.com/yunus6141/jsvelox

// 01

jsvelox Nedir?

jsvelox, Claude, OpenAI ve Gemini gibi yapay zeka sağlayıcılarını birkaç satır kodla projenize entegre etmenizi sağlayan hafif bir JavaScript kutuphanesidir. Kullanılmayan özellikler için bile kaynak tüketen ağır yapay zeka agent platformlarının aksine, jsvelox basit bir prensiple çalışır: sadece ihtiyacınız olanı kullanın.

Görsel pipeline oluşturucular veya karmaşık sunucu kurulumu olmadan yapay zeka entegrasyonu isteyen geliştiriciler için tasarlanmıştır. Yukleyin, birkaç satır yazın ve yapay zeka asistanınız hazır.

// 02

jsvelox Hangi Sorunları Çözüyor?

Cok Karmaşık	Yapay zeka agent platformları görsel arayüzler, karmaşık yapılandırmalar ve diğer öğrenme eğrileri gerektirir.	✓ jsvelox basit JavaScript nesneleri kullanır — sür-bırak yok, görsel editorler yok.
Cok Ağır	n8n gibi platformlar kullanılmayan node'lar için bile RAM tüketir.	✓ jsvelox sadece gerçekten çağırdığınız şeyler için kaynak kullanır.
Cok Fazla Dosya	Yapay zekayı entegre etmek genellikle birden fazla API istemcisi yönetimi demektir.	✓ Bir kütüphane, uc yapay zeka sağlayıcısı, birleşik API.
Ozelleştirmesi Zor	Yapay zekayı kendi verilerinizle eğitmek genellikle karmaşık kurulumlar gerektirir.	✓ systemPrompt ve context aynı konfigürasyon nesnesinde yapay zekayı eğitmenizi sağlar.

// 03

Kod Yapısı

jsvelox tek bir sinif etrafında inşa edilmistir: VeloxAI. Nasıl çalıştığı:

Zorunlu Konfigürasyon

```
import VeloxAI from 'jsvelox'

const ai = new VeloxAI({
  provider: 'gemini', // Zorunlu: claude | openai | gemini
  apiKey: 'anahtarınız', // Zorunlu: API anahtarınız
  memoryLimit: 10, // Opsiyonel: max kac mesaj hatirlansin
  timeout: 15000 // Opsiyonel: kac ms'de zaman asimi
})

const cevap = await ai.send('Merhaba!')
console.log(cevap)
```

Yapay Zeka Egitim Konfigurasyonu

```
const ai = new VeloxAI({
  provider: 'gemini',
  apiKey: 'anahtarınız',
  systemPrompt: 'Sen bir magaza asistanisin. Türkçe cevap ver.',
  context: [
    { urun: 'Urun A', fiyat: '299 TL', fayda: 'Enerji verir' },
    { urun: 'Urun B', fiyat: '499 TL', fayda: 'Uykuyu düzeltir' }
  ]
})
```

// 04

Mevcut Metodlar

Method	Description
<code>send(mesaj)</code>	Mesaj gönder ve cevap al
<code>sendWithRetry(mesaj, n)</code>	Hata durumunda otomatik tekrar dene (varsayılan: 3 kez)
<code>onSend(btn, input, res)</code>	Buton/input/cevap alanını tek satırda bağla
<code>getMemory()</code>	Konusma geçmişini göster
<code>clearMemory()</code>	Konusma geçmişini temizle

Yaygin Hatalar ve Cozumler

HATA	API key zorunlu!
Cause:	VeloxAI olusturulurken apiKey verilmemis.
Fix:	Her zaman konfigurasyon nesnesinde apiKey girin: <pre>new VeloxAI({ provider: 'gemini', apiKey: 'anahtariniz' })</pre>
HATA	Provider zorunlu!
Cause:	Hangi yapay zeka saglayicisinin kullanilacagi belirtilmemis.
Fix:	Konfigurasyona provider ekleyin: <pre>new VeloxAI({ provider: 'gemini', apiKey: 'anahtar' })</pre>
HATA	Gecersiz provider!
Cause:	Desteklenmeyen bir saglayici adi yazdiniz. Ornek: provider: 'gpt'
Fix:	Sadece su degerleri kullanin: claude openai gemini Not: jsvelox buyuk harfleri otomatik kucuk harfe cevirir.
HATA	Mesaj bos olamaz!
Cause:	ai.send() bos string veya boslukla cagirildi.
Fix:	Gondermeden once mesaji kontrol edin: <pre>if (mesaj.trim()) { await ai.send(mesaj) }</pre>
UYARI	Zaman asimi! AI cevap vermedi.
Cause:	Yapay zeka belirlenen sure icinde (varsayilan 30 saniye) cevap vermedi.
Fix:	Timeout artirin veya sendWithRetry kullanin: <pre>new VeloxAI({ timeout: 60000 }) await ai.sendWithRetry(mesaj, 3)</pre>
HATA	Claude hata: invalid x-api-key
Cause:	API anahtariniz yanlis veya suresi dolmus.

Fix:	Gecerli anahtar alın: - Claude: console.anthropic.com - OpenAI: platform.openai.com - Gemini: aistudio.google.com
UYARI	Quota exceeded (Kota asimi)
Cause:	Gunluk ucretsiz API kotanizi kullandınız.
Fix:	Kota sifirlaniyor (genellikle gece yarisi). sendWithRetry gecici hatalar icin kullanilabilir.
HATA	onSend: Belirtilen elementler bulunamadi!
Cause:	onSend()'e gecilen CSS seciciler hicbir HTML elementine eslesmiyor.
Fix:	Secicilerinizi kontrol edin: ai.onSend('#buton', '#input', '#cevap') Bu ID'lerin HTML'inizde mevcut oldugunu dogrulayin.
UYARI	CORS Hatasi (tarayici)
Cause:	API dogrudan tarayicidan cagiriliyor. Tarayicilar cross-origin istekleri engeller.
Fix:	jsvelox'u Node.js backend'inde kullanin, dogrudan tarayicida degil. v3.0'da backend proxy destegi gelecek.

// 06

try-catch ile Hata Yonetimi

```
try {
  const cevap = await ai.send('Merhaba!')
  console.log(cevap)
} catch (hata) {
  if (hata.message.includes('Zaman asimi')) {
    console.log('Yapay zeka cok uzun surdu, tekrar deneyin')
  } else if (hata.message.includes('api-key')) {
    console.log('API anahtarinizi kontrol edin')
  } else {
    console.log('Hata:', hata.message)
  }
}
```

// 07

jsvelox En Iyi Ne Zaman Calisir?

- ✓ Node.js backend uygulamalari
- ✓ Express.js veya Fastify sunuculari
- ✓ Next.js API rotalari
- ✓ CLI araclari ve scriptler
- ✓ Node.js 18+ üzerinde calisan herhangi bir JavaScript/TypeScript projesi

jsvelox Ne Zaman Calismaz?

- ✗ Dogrudan tarayici kullanimi (CORS kisitlamasi) — v3.0'da gelecek
- ✗ Deno veya Bun ortamlari (test edilmedi)
- ✗ Node.js 18 altı surumler (fetch yerlesik degil)

// 08

TypeScript Kullanimi

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

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

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

