

jsvelox - 完全ガイド

jsveloxについて知る必要があるすべて
v2.2.0 - コードネーム: Quagga

github.com/yunus6141/jsvelox

// 01

jsveloxとは？

jsveloxは、Claude、OpenAI、GeminiなどのAIプロバイダーを数行のコードでプロジェクトに統合できる軽量JavaScriptライブラリです。未使用の機能のためにもリソースを消費する重いAIエージェントプラットフォームとは異なり、jsveloxはシンプルな原則に従います：必要なものだけ使用する。

ビジュアルパイプラインビルダーや複雑なサーバーセットアップなしでAI統合を望む開発者のために設計されています。

// 02

jsveloxが解決する問題

複雑すぎる	AIエージェントプラットフォームはビジュアルインターフェースと急な学習曲線が必要です。	☑ jsveloxはシンプルなJavaScriptオブジェクトを使用します。
重すぎる	n8nのようなプラットフォームは未使用のノードでもRAMを消費します。	☑ jsveloxは実際に呼び出したものにのみリソースを使用します。
ファイルが多すぎる	AI統合は通常、複数のAPIクライアントの管理を意味します。	☑ 1つのライブラリ、3つのAIプロバイダー、統一API。
カスタマイズが難しい	独自データでAIをトレーニングするには複雑なセットアップが必要です。	☑ systemPromptとcontextで同じ設定オブジェクトでAIをトレーニングできます。

// 03

コード構造

jsveloxは1つのクラスを中心に構築されています：VeloxAI。動作方法：

必須設定

```
import VeloxAI from 'jsvelox'

const ai = new VeloxAI({
  provider: 'gemini', // 〇〇: claude | openai | gemini
  apiKey: '〇〇〇〇〇〇', // 〇〇: API〇〇
  memoryLimit: 10,      // 〇〇〇〇〇: 〇〇〇〇〇〇〇〇〇〇〇〇
  timeout: 15000        // 〇〇〇〇〇: 〇〇〇〇〇〇〇〇〇〇ms
})

const reply = await ai.send('〇〇〇〇〇〇')
console.log(reply)
```

AIトレーニング設定

```
const ai = new VeloxAI({
  provider: 'gemini',
  apiKey: '〇〇〇〇〇〇',
  systemPrompt: '〇〇〇〇〇〇〇〇〇〇〇〇〇〇〇〇〇〇〇〇〇〇〇〇〇〇〇〇',
  context: [
    { 〇〇〇: '〇〇〇A', 〇〇: '2900〇', 〇〇〇〇: '〇〇〇〇〇〇〇' },
    { 〇〇〇: '〇〇〇B', 〇〇: '4900〇', 〇〇〇〇: '〇〇〇〇〇' }
  ]
})
```

// 04

利用可能なメソッド

Method	Description
send(message)	メッセージを送信して応答を受信
sendWithRetry(msg, n)	失敗時に自動リトライ（デフォルト：3回）
onSend(btn, input, res)	ボタン/入力/応答を1行でバインド
getMemory()	完全な会話履歴を取得
clearMemory()	会話履歴をクリア

一般的なエラーと解決策

エラー	APIキーが必要です！
Cause:	apiKeyなしでVeloxAIが作成されました。
Fix:	<pre>■■apiKey■■■■■■■■■■ new VeloxAI({ provider: 'gemini', apiKey: '■■' })</pre>
エラー	プロバイダーが必要です！
Cause:	AIプロバイダーが指定されていません。
Fix:	<pre>provider■■■■ new VeloxAI({ provider: 'gemini', apiKey: '■■' })</pre>
エラー	無効なプロバイダー！
Cause:	サポートされていないプロバイダー名。例：provider: 'gpt'
Fix:	<pre>■■■■■■■■■claude openai gemini ■■</pre>
エラー	メッセージは空にできません！
Cause:	ai.send()が空文字列で呼び出されました。
Fix:	<pre>■■■■■■■■■■■■■■■■ if (msg.trim()) { await ai.send(msg) }</pre>
警告	タイムアウト！
Cause:	AIが制限時間内に応答しませんでした。
Fix:	<pre>■■■■■■■■■■■■■■■■sendWithRetry■■■■■■■■■■■■■■■■</pre>
エラー	無効なAPIキー
Cause:	APIキーが間違っているか期限切れです。
Fix:	<pre>■■■■■■■■■■console.anthropic.com■■platform.openai.com■■aistudio.google .com</pre>
警告	クォータ超過

[illegible]

// 06

try-catchによるエラー処理

```
try {
  const reply = await ai.send('■■■■■■■')
  console.log(reply)
} catch (err) {
  if (err.message.includes('Timeout')) {
    console.log('AI■■■■■■■■■')
  } else {
    console.log('■■■:', err.message)
  }
}
```

// 07

jsveloxが最も効果的な場合

- ✓ Node.jsバックエンドアプリケーション
- ✓ Express.jsまたはFastifyサーバー
- ✓ Next.js APIルート
- ✓ CLIツールとスクリプト
- ✓ Node.js 18+で動作するJavaScript/TypeScriptプロジェクト

jsveloxが機能しない場合

- ☒ ブラウザからの直接使用（CORS制限） - v3.0で対応予定
- ☒ DenoまたはBun環境（未テスト）
- ☒ Node.js 18未満のバージョン

// 08

TypeScriptの使用

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

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

const ai = new VeloxAI(config)
const reply: string = await ai.send('■■■■■■■')
```

