Vercel AI SDK

The TypeScript Toolkit for Building AI Applications

Executive Summary

The Vercel AI SDK is a free, open-source TypeScript library with over 20 million monthly downloads that provides a unified API for building AI applications across any provider and framework. Its core value proposition is eliminating provider lock-in—developers can switch between OpenAI, Anthropic, Google, and 30+ other providers with minimal code changes while getting production-ready streaming, tool calling, and structured output out of the box.

The latest AI SDK 6 (December 2025) introduced agent abstractions, full MCP support, and DevTools for debugging complex AI workflows.

Architecture Overview

The SDK is architected as two complementary libraries:

AI SDK Core

Provides server-side functions for text generation, structured objects, tool calling, and agent workflows:

- generateText / streamText – Primary text generation functions
- generateObject / streamObject – Structured data generation
- embed / embedMany – Vector embeddings for RAG applications
- Tool definitions with Zod schema validation

AI SDK UI

Offers framework-agnostic hooks for building responsive chat interfaces:

- useChat – Full-featured chat interface management
- useCompletion – Single-turn completion handling
- useObject – Streaming structured JSON objects
- Supports React, Vue, Svelte, and SolidJS

Supported Providers

The SDK supports 30+ first-party providers through official packages:

Provider	Package	Notable Capabilities

OpenAI	@ai-sdk/openai	GPT-4o, image generation, predicted outputs
Anthropic	@ai-sdk/anthropic	Claude 4, PDF support, computer use, MCP tools
Google	@ai-sdk/google	Gemini 2.0, Maps integration, dynamic retrieval
Amazon Bedrock	@ai-sdk/amazon-bedrock	All Bedrock models, reasoning support
xAI Grok	@ai-sdk/xai	Web search, X/Twitter search, code execution
Mistral	@ai-sdk/mistral	PDF support, tool streaming
Together.ai	@ai-sdk/togetherai	Image generation, open models
DeepSeek	@ai-sdk/deepseek	Reasoning models, tool streaming

Additional providers include ElevenLabs and LMNT for text-to-speech, AssemblyAI and Deepgram for transcription, and Fal AI and Black Forest Labs for image generation. Community providers extend support to Ollama for local models, Cloudflare Workers AI, OpenRouter, and Portkey for routing.

Core APIs

Text Generation

generateText and streamText are the primary text generation functions. Use generateText for non-interactive tasks like email drafting or agent workflows, and streamText for real-time chat interfaces.

Both functions return rich metadata:

- text – The generated text content
- toolCalls – Any tool invocations made
- toolResults – Results from tool executions
- finishReason – Why generation stopped
- usage – Token counts for billing/monitoring
- reasoning – For reasoning models like o1

Embeddings

embed and embedMany generate vector embeddings for semantic search and RAG applications. They support OpenAI's text-embedding-3 models, Google's gemini-embedding-001, Cohere, Mistral, and Amazon Titan.

React Hooks for Chat Interfaces

useChat Hook

The primary hook for building conversational interfaces. It manages message state, handles streaming, processes tool calls, and provides status tracking.

Returns:

- messages – Array of UIMessage objects with id, role, parts, status
- status – Current state: "submitted", "streaming", "ready", or "error"
- sendMessage – Function to send new messages
- stop – Function to abort streaming
- regenerate – Function to regenerate the last response
- setMessages – Function to programmatically update messages

Additional Hooks

useCompletion: Handles single-turn completions with input management and form submission

useObject: (Experimental) Streams structured JSON objects as they generate for real-time updates

Framework Support

Framework	Package	Features
Next.js App Router	@ai-sdk/react	RSC, streamUI for generative UI, Server Actions
Next.js Pages Router	@ai-sdk/react	API routes, streaming
SvelteKit	@ai-sdk/svelte	Full streaming, SSR
Nuxt/Vue	@ai-sdk/vue	Composition API hooks
SolidStart	@ai-sdk/solid	Hooks support
Expo	@ai-sdk/react	Mobile AI applications

Runtime support includes Node.js 18+, Edge Runtime (optimized for Vercel), Deno, Cloudflare Workers, and Bun.

Streaming Capabilities

The SDK implements efficient SSE-based streaming protocols:

Data Stream Protocol (Default)

Uses Server-Sent Events to transmit text, tool calls, reasoning, sources, and custom data with keep-alive pings and reconnection support.

Stream part types include:

- text-start/delta/end – Text content streaming
- reasoning-start/delta/end – Reasoning content for o1-style models
- tool-input-start/delta/available – Tool invocation streaming
- tool-output-available – Tool result delivery
- source-url – Citation/source references
- file – File content streaming
- data-* – Custom data types

Backpressure is automatic—streamText only generates tokens as they are requested, preventing memory issues with slow consumers.

Tool Calling & Agents

Tools are defined with the tool() helper using Zod schemas for type-safe inputs. The SDK supports multi-step tool calling with configurable stop conditions.

Stop Conditions

- stepCountIs(n) – Stop after n steps
- hasToolCall("name") – Stop when specific tool is called
- Custom conditions via callback functions

Tool Options

- toolChoice – Control tool selection: "auto", "required", "none", or specific tool
- Lifecycle hooks – onInputStart, onInputDelta, onInputAvailable
- needsApproval – Enable human-in-the-loop workflows

ToolLoopAgent (AI SDK 6)

A production-ready agent class that handles complete tool execution loops. Configure with model, system prompt, tools, and stop conditions for reusable agent patterns.

Structured Output

The Output API (AI SDK 6) generates type-safe structured data with schema validation:

- Output.object() – Generate structured objects with Zod schemas
- Output.array() – Generate typed arrays
- Output.choice() – Classification/enum selection
- Output.json() – Unstructured JSON output

Streaming structured output uses partialOutputStream to display partially-generated objects in real-time. Critically, structured output now works with multi-step tool calling—generate text with tools, then produce a structured result at the end, all in one request.

AI SDK 6 Features (December 2025)

Major capabilities introduced:

- Agent Abstraction – ToolLoopAgent class for reusable agents with model, instructions, and tools
- Tool Execution Approval – needsApproval flag for human-in-the-loop control
- DevTools – @ai-sdk/devtools package for debugging multi-step agent flows

- Full MCP Support – Stable Model Context Protocol in @ai-sdk/mcp with OAuth, resources, prompts
- Reranking – rerank() function supporting Cohere, Bedrock, Together.ai
- Image Editing – generateImage() now supports image-to-image operations
- Provider Tools – Anthropic memory/code execution, OpenAI shell/MCP tools, Google Maps, xAI web search

Getting Started

Installation

```
pnpm add ai @ai-sdk/react zod
```

Environment Setup

Set AI_GATEWAY_API_KEY in .env.local for the Vercel AI Gateway (recommended), or use provider-specific keys like OPENAI_API_KEY with direct provider packages.

Basic Usage Pattern

1. Create an API route to handle chat requests

2. Use streamText() with your chosen model

3. Return result.toUIMessageStreamResponse()

4. Build your UI with useChat() hook

Conclusion

The Vercel AI SDK provides a production-grade foundation for AI applications with three key advantages:

- Provider Abstraction – Switch models freely without code changes
- Streaming Infrastructure – Handles complexity automatically with efficient protocols
- Type-Safe Tooling – Zod schemas throughout for compile-time safety

The AI SDK 6 release positions it strongly for agentic applications with ToolLoopAgent, MCP integration, and DevTools. For teams building AI features, it eliminates months of infrastructure work while maintaining flexibility to use any model from any provider.

Resources

- Documentation: https://ai-sdk.dev
- GitHub: https://github.com/vercel/ai
- AI SDK 6 Announcement: https://vercel.com/blog/ai-sdk-6