LangChain Docs

Releases

Changelog

📋 Copy page   ⌄   📶

Log of updates and improvements to our JavaScript/TypeScript packages

Subscribe: Our changelog includes an **RSS feed** that can integrate with **Slack**, **email**, Discord bots like **Readybot** or **RSS Feeds to Discord Bot**, and other subscription tools.

Jan 14, 2026

langgraph

v1.1.0

`@langchain/langgraph`

Introducing **StateSchema** - a cleaner, library-agnostic way to define graph state that works with any **Standard Schema**-compliant validation library.

Standard JSON Schema support

LangGraph now supports **Standard JSON Schema**, an open specification implemented by Zod 4, Valibot, ArkType, and other schema libraries. This means you can use your preferred validation library without lock-in:

LangChain Docs

```
  messages: MessagesValue,
  currentStep: z.string(),
  count: z.number().default(0),
  history: new ReducedValue(
    z.array(z.string()).default(() => []),
    {
      inputSchema: z.string(),
      reducer: (current, next) => [...current, next],
    }
  ),
});

// Type-safe state and update types
type State = typeof AgentState.State;
type Update = typeof AgentState.Update;

const graph = new StateGraph(AgentState)
  .addNode("agent", (state) => ({ count: state.count + 1 }))
  .addEdge(START, "agent")
  .addEdge("agent", END)
  .compile();
```

New state value primitives

ReducedValue: Define fields with custom reducers for accumulating values. Supports separate input and output schemas for type-safe reducer inputs.

UntrackedValue: Define transient state that exists during execution but is never checkpointed - useful for database connections, caches, or runtime-only configuration.

MessagesValue: A prebuilt `ReducedValue` for chat messages with the standard messages reducer.

Type helper exports

exported type utilities for typing functions outside the graph builder:

LangChain Docs

```
const myNode: GraphNode<typeof AgentState> = (state, config) => {
  return { count: state.count + 1 };
};

// Use schema type helpers directly
const processState = (state: typeof AgentState.State) => {
  console.log(state.count);
};
```

The existing `Annotation` and zod-based API continues to work unchanged - `StateSchema` is an additional option for those who prefer schema-first definitions.

Learn more about StateSchema

See the full documentation for defining graph state with StateSchema, ReducedValue, and UntrackedValue.

Learn about type utilities

Use GraphNode and ConditionalEdgeRouter to type functions outside the graph builder.

Dec 12, 2025

langchain @langchain/openai @langchain/anthropic @langchain/ollama @langchain/community
@langchain/xai @langchain/tavily @langchain/mongodb @langchain/mcp-adapters
gchain/google-common @langchain/core

v1.2.0

LangChain Docs

☰ Releases › **Changelog**

...for structured output.

@langchain/openai

New provider built-in tools: Support for <u>file search</u>, <u>web search</u>, <u>code interpreter</u>, <u>image generation</u>, <u>computer use</u>, <u>shell</u>, and <u>MCP connector</u> tools.

Content moderation: New `moderateContent` option on `ChatOpenAI` for detecting and handling unsafe content.

Prefer responses API for GPT-5.2 Pro model.

v1.3.0

@langchain/anthropic

New provider built-in tools: Support for <u>text editor</u>, <u>web fetch</u>, <u>computer use</u>, <u>tool search</u>, and <u>MCP toolset</u> tools.

Exposed `ChatAnthropicInput` type for improved type safety.

v1.1.0

@langchain/ollama

Native structured outputs: Added support for native structured output via `withStructuredOutput`.

Support for custom `baseUrl` configuration.

v1.0.0

@langchain/community

Jira document loader updated to use v3 API.

LanceDB: Added `similaritySearch()` and `similaritySearchWithScore()` support.

Elasticsearch hybrid search support.

New `GoogleCalendarDeleteTool`.

![icon] arious bug fixes for LlamaCppEmbeddings, PrismaVectorStore, IBM WatsonX, and security improvements.

LangChain Docs

@langchain/mcp-adapters: Added `onConnectionError` option.

@langchain/google-common: `jsonSchema` method support in `withStructuredOutput`.

@langchain/core: Security fixes, better subgraph nesting in Mermaid graphs, UUID7 for run IDs.

v1.1.0

Model profiles: Chat models now expose supported features and capabilities through a `.profile` getter. These data are derived from **models.dev**, an open source project providing model capability data.

Model retry middleware: New middleware for automatically retrying failed model calls with configurable exponential backoff, improving agent reliability.

Content moderation middleware: OpenAI content moderation middleware for detecting and handling unsafe content in agent interactions. Supports checking user input, model output, and tool results.

Summarization middleware: Updated to support flexible trigger points using model profiles for context-aware summarization.

Structured output: `ProviderStrategy` support (native structured output) can now be inferred from model profiles.

`SystemMessage` for `createAgent`: Support for passing `SystemMessage` instances directly to `createAgent`'s `systemPrompt` parameter and a new `concat` method for extending system messages. Enables advanced features like cache control and structured content blocks.

Dynamic system prompt middleware: Return values from `dynamicSystemPromptMiddleware` are now purely additive. When returning a `SystemMessage` or `string`, they are merged with existing system messages rather than replacing them, making it easier to compose multiple middleware that modify the prompt.

Compatibility improvements: Fixed error handling for Zod v4 validation errors in structured output and tool schemas, ensuring detailed error messages are properly displayed.

Oct 20, 2025

langchain langgraph

v1.0.0

langchain

Release notes

Migration guide

LangChain Docs

Migration guide

> If you encounter any issues or have feedback, please **open an issue** so we can improve. To view v0.x documentation, **go to the archived content**.

Edit this page on GitHub or **file an issue**.

💡 **Connect these docs** to Claude, VSCode, and more via MCP for real-time answers.

Was this page helpful?   👍 Yes   👎 No

LangChain Docs

🔗

LangChain Academy

Blog

Trust Center

Powered by mintlify

LangChain Docs