Installation | Langchain

Skip to main content LangChainDocsUse casesIntegrationsAPIMoreCommunityTutorialsContributingAlso by LangChainChat our docsLangSmithLangChain HubLangServePython DocsSearchCTRLKGet startedIntroductionInstallationQuickstartLangChain Expression LanguageInterfaceHow toCookbookWhy use LCEL?LangChain Expression Language (LCEL)ModulesModel I/ORetrievalChainsMemoryAgentsCallbacksModulesSecurityGuidesEcosystemGet startedInstallationOn this pageInstallationinfoUpdating from <0.0.52? See this section for instructions.Supported EnvironmentsLangChain is written in TypeScript and can be used in:Node.js (ESM and CommonJS) - 18.x, 19.x, 20.xCloudflare WorkersVercel / Next.js (Browser, Serverless and Edge functions)Supabase Edge FunctionsBrowserDenoBunInstallationTo get started, install LangChain with the following command:npmYarnpnpmnpm install -S langchainyarn add langchainpnpm add langchainTypeScriptLangChain is written in TypeScript and provides type definitions for all of its public APIs.Loading the libraryESMLangChain provides an ESM build targeting Node.js environments. You can import it using the following syntax:import { OpenAI } from "langchain/llms/openai";If you are using TypeScript in an ESM project we suggest updating your tsconfig.json to include the following:tsconfig.json{   "compilerOptions": {   ...   "target": "ES2020", // or higher   "module": "nodenext",   }}CommonJSLangChain provides a CommonJS build targeting Node.js environments. You can import it using the following syntax:const { OpenAI } = require("langchain/llms/openai");Cloudflare WorkersLangChain can be used in Cloudflare Workers. You can import it using the following syntax:import { OpenAI } from "langchain/llms/openai";Vercel / Next.jsLangChain can be used in Vercel / Next.js. We support using LangChain in frontend components, in Serverless functions and in Edge functions. You can import it using the following syntax:import { OpenAI } from "langchain/llms/openai";Deno / Supabase Edge FunctionsLangChain can be used in Deno / Supabase Edge Functions. You can import it using the following syntax:import { OpenAI } from "https://esm.sh/langchain/llms/openai";We recommend looking at our Supabase Template for an example of how to use LangChain in Supabase Edge Functions.BrowserLangChain can be used in the browser. In our CI we test bundling LangChain with Webpack and Vite, but other bundlers should work too. You can import it using the following syntax:import { OpenAI } from "langchain/llms/openai";Updating from <0.0.52If you are updating from a version of LangChain prior to 0.0.52, you will need to update your imports to use the new path structure.For example, if you were previously doingimport { OpenAI } from "langchain/llms";you will now need to doimport { OpenAI } from "langchain/llms/openai";This applies to all imports from the following 6 modules, which have been split into submodules for each integration. The combined modules are deprecated, do not work outside of Node.js, and will be removed in a future version.If you were using langchain/llms, see LLMs for updated import paths.If you were using langchain/chat_models, see Chat Models for updated import paths.If you were using langchain/embeddings, see Embeddings for updated import paths.If you were using langchain/vectorstores, see Vector Stores for updated import paths.If you were using langchain/document_loaders, see Document Loaders for updated import paths.If you were using langchain/retrievers, see Retrievers for updated import paths.Other modules are not affected by this change, and you can continue to import them from the same path.Additionally, there are some breaking changes that were needed to support new environments:import { Calculator } from "langchain/tools"; now moved toimport { Calculator } from "langchain/tools/calculator";import { loadLLM } from "langchain/llms"; now moved toimport { loadLLM } from "langchain/llms/load";import { loadAgent } from "langchain/agents"; now moved toimport { loadAgent } from "langchain/agents/load";import { loadPrompt } from "langchain/prompts"; now moved toimport { loadPrompt } from "langchain/prompts/load";import { loadChain } from "langchain/chains"; now moved toimport { loadChain } from "langchain/chains/load";Unsupported: Node.js 16We do not support Node.js 16, but if you still want to run LangChain on Node.js 16, you will need to follow the instructions in this section. We do not guarantee that these instructions will continue to work in the future.You will have to make fetch available globally, either:run your application with NODE_OPTIONS='--experimental-fetch' node ..., orinstall node-fetch and follow the instructions hereYou'll also need to polyfill ReadableStream by installing:npmYarnpnpmnpm i web-streams-polyfillyarn add web-streams-polyfillpnpm add web-streams-polyfillAnd then adding it to the global namespace in your main entrypoint:import "web-streams-polyfill/es6";Additionally you'll have to polyfill structuredClone, eg. by installing core-js and following the instructions here.If you are running Node.js 18+, you do not need to do anything.PreviousIntroductionNextQuickstartSupported EnvironmentsInstallationTypeScriptLoading the libraryESMCommonJSCloudflare WorkersVercel / Next.jsDeno / Supabase Edge FunctionsBrowserUpdating from <0.0.52Unsupported: Node.js 16CommunityDiscordTwitterGitHubPythonJS/TSMoreHomepageBlogCopyright © 2023 LangChain, Inc.