laiso

2025-01-11

MCPクライアントアプリを作ってコマンドラインでエージェントを走らせよう

なぜ MCP クライアント側開発の解説が必要か

Model Context Protocol（MCP） は、さまざまな AI アプリケーションや LLM（Large Language Model）を外部ツール・リソース・ワークフローに接続するためのプロトコルです。これはサーバーとクライアントの 2 つの側面があるのですが、現在 Web 検索などで見つかる情報は主に「MCPサーバーの呼び出し方」と「(呼び出される）MCPサーバーの作り方」というテーマが中心になっています。しかし実際に MCP サーバーを呼び出せるのは、Claude アプリや Zed などのエディタ*1、VS Code 拡張の Cline*2 といった、すでにクライアント（ホストアプリケーション）を実装している限られたアプリケーションに限られます。
そのため、サーバーを実装するだけでは、現状対応しているアプリ(多くはCalaudeアプリ)を活用することにとどまります。MCPサーバー連携機能を自分たちのアプリへ実装し、ユーザーへ提供する方法はあまり知られていません。私はその「強化される側」であるクライアント側を独自に作りたいと考えており、「どうやってサーバーを利用する新しいアプリを開発するのか」という方法について知る必要がありました。

www.anthropic.com

MCP クライアントアプリを作るには？

MCP では標準出力（Stdio）や SSE（Server-Sent Events）を使った JSON-RPC メッセージングによって外部プログラムと通信します。現在はローカル環境での利用が中心ですが、今後はリモート呼び出しにも対応する予定です*3。SDK（ライブラリ）としてはオフィシャルの Python 版、もしくは TypeScript 版が存在します。Python 版の方が現状では機能面の充実度が高い傾向にあります。しかしブラウザやNext.jsアプリの裏側から使うケースを見込んで個人的にはTypeScript 版を選択しています。

```
import { Client } from "@modelcontextprotocol/sdk/client/index.js";
import { StdioClientTransport } from
"@modelcontextprotocol/sdk/client/stdio.js";
import { CallToolRequestSchema } from
"@modelcontextprotocol/sdk/types.js";

// クライアントの設定
const client = new Client(
  {
    name: "example-client",
    version: "1.0.0",
  },
  {
    capabilities: {
      tools: {}, // ツールを使用するためのcapabilityを設定
    },
  }
);

// stdioトランスポートの設定
```

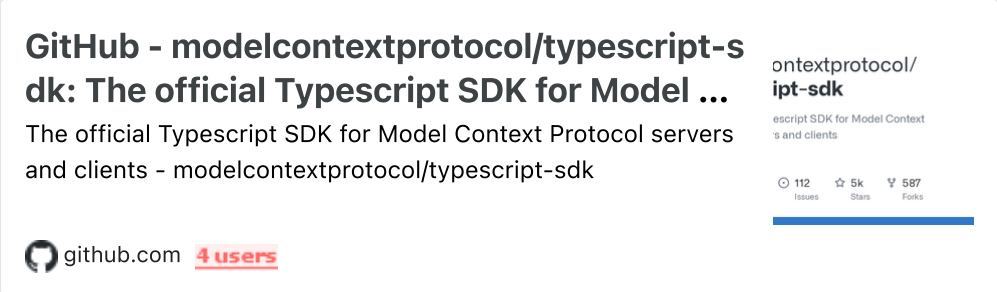
```
const transport = new StdioClientTransport({
  command: "./path/to/your/mcp-server", // MCPサーバーの実行パス
});

async function main() {
    // クライアントをサーバーに接続
    await client.connect(transport);
    console.log("Connected to MCP server");

  // ツールを呼び出す
  try {
    const callToolResult = await client.request(
      {
        method: "tools/call",
        params: {
          name: "your-tool-name", // 呼び出すツール名
          arguments: {
            // ツールに渡す引数
            input1: "value1",
            input2: 123,
          },
        },
      },
      CallToolRequestSchema // 型定義スキーマ
    );

    console.log("Tool call result:", callToolResult);

  } catch (error) {
    console.error("Error calling tool:", error);
  } finally {
    // クライアントを閉じる
    await client.close();
  }
}
```



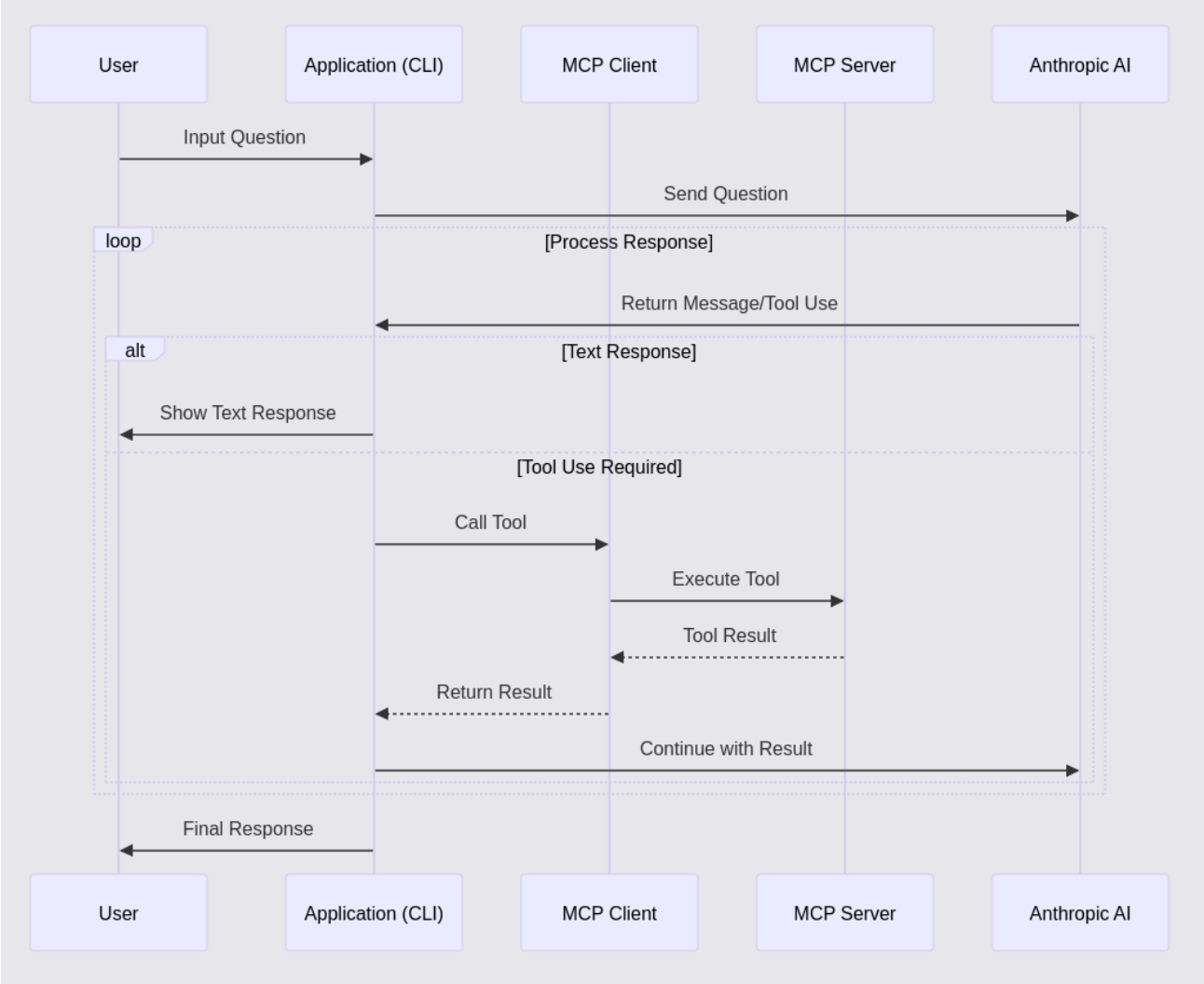
github.com

MCP の基本的な概念としては、大きく **Resources**、**Tools**、**Prompts**、**Sampling**、**Roots** に分かれています。はじめて実装する場合は、とくに Tools の使い方を理解するのがおすすめです。たとえば MCP サーバーからツール一覧を取得し、必要なツールを呼び出すといった流れが基本になります。

クライアント側での具体的な手順としては、まずアプリケーション内に接続先サーバー情報を埋め込む（または設定ファイルで指定する）などして MCP サーバーを登録し、SDKのClientモジュールを準備したうえで、ツール一覧を取得します。取得したツール一覧を LLM 側に教える際は、対応する LLM の仕様に合わせた形式でツールの名前や機能をプロンプトに埋め込みます。モデルのプロバイダによってAPI仕様が違うので、ここでは、Anthropic の API を前提に話を進めます。

docs.anthropic.com

Anthropic の APIにテキスト指示を送信した際、レスポンスとして "TOOL_USE" が返ってくるケースがあります。これは「指定されたツールを使って処理するように」という合図なので、実際にアプリの内部で SDKを使ってMCP サーバーのツールを呼び出し、得られた結果を再度 LLM に返すという流れになります。
このようにしてクライアント側を実装することで、MCP サーバーと連携した機能を新しく作成し、LLM を活用した高度な処理をアプリ内で自在に行えるようになります。



クライアントアプリのアーキテクチャ

どんなクライアントアプリがつくれるか？

MCP のクライアント実装は、さまざまな形で応用できます。たとえば、既存のウェブアプリに組み込んだチャット機能が外部の MCP サーバーを呼び出せるようにすれば、サーバー上のツールを使って高度なデータ処理を行ったり、FAQ を自動生成したりすることも可能になります。また、VS Code だけでなく、別のエディタ向けにプラグインを開発して MCP を呼び出す仕組みを組み込むことも考えられます。さらに、Python や JavaScript（Electron など）を使ってデスクトップアプリを作り、ローカル環境で MCP サーバーと連携させることで、専用の生産性向上ツールやコマンドラインユーティリティを開発するといった可能性も広がるでしょう。

MCPの公式サイトには標準で提供されているMCPサーバーのリストやクライアントアプリを作る開発者向けの情報があります。

For Client Developers - Model Context Protocol
Get started building your own client that can integrate with all MCP servers.
modelcontextprotocol.io 3 users

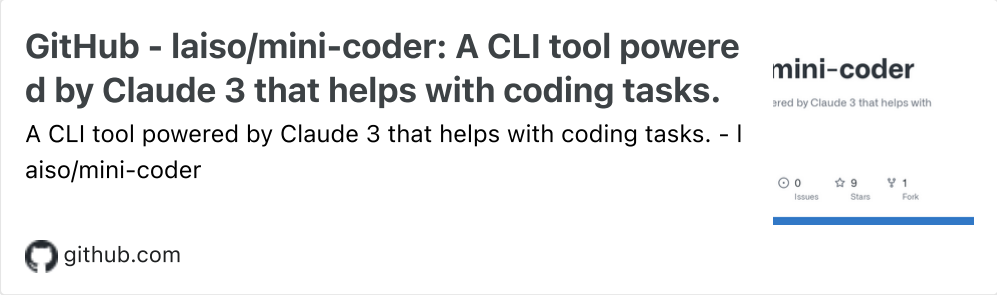
modelcontextprotocol.io

ミニコーディングエージェントを作ってみた

MCP におけるファイルシステム（Filesystem）サーバー*4を組み込んで、特定ディレクトリに対する指示（タスク）を実行するコマンドラインツールを作成しました。これは Cline のようなエージェントをヘッドレスにサーバー上で動かしたり、Devin のようにサービス化したりするといったアイデアを目標としています。開発工程については Zenn で複数回にわたり記事としてまとめてありますので、興味があればぜひ参考にしてください。

1. TypeScriptでMCPサーバーのtool呼び出しをする
2. MCPサーバーのtool呼び出しをLLMにトリガーしてもらう
3. MCPを使って160行のTypeScriptでミニコーディングエージェントを作る

リポジトリはこちらです。



github.com

なお、実際に動かす際には API の利用料金がかかります。金銭的な負担が気になる場合は実行をせず、手順だけを理解してMCP クライアントの作り方を学べます。これを参考に、皆さま独自のクライアントを開発していただければと思います。

*1: The Context Outside the Code

*2: ClineとAIコーディングツールの現状 - laiso

*3: Roadmap - Model Context Protocol

*4: Example Servers - Model Context Protocol

laiso (id:laiso) 99日前　読者になる

26

42

« NotebookLMでポッドキャスト生成して MacWh…

ClineとAIコーディングツールの現状 »

Author

laiso (id:laiso)

https://blog.lai.so/ に移転済み

読者になる 434

@laisoをフォロー