

MCP の通信内容を見てみる

奥村 義和

本章では、ユーザがアプリケーションを通して、どのようにMCP (Model Context Protocol) サーバやLLM (Large Language Models) とやりとりしているかを見てみます。具体的にはVisual Studio Code (以降、VS Code) で動作するAIアシスタントの拡張機能Cline^{注1}を利用します。Clineは開発者がやりたいことを対話形式で指示するだけで、タスクの計画から実行までを自律的に行うので、ビギナにもお勧めです。

Clineがユーザからの指示を受け、MCPサーバやLLMを呼び出します(図1)。

VS Codeで動くAIアシスタント Clineを使ってみる

● チャットでプログラムを作ってみる

Clineにはチャット欄が用意されており、「pythonでhello worldを出力するプログラムを作成して」などと入力すると、プログラムを作成、実行してくれます。

複雑なプログラムを作成する際は、Planモードで仕様と一緒に検討し、仕様が固まったらActモードに切り替えてプログラムを作成します。

注1: <https://cline.bot>

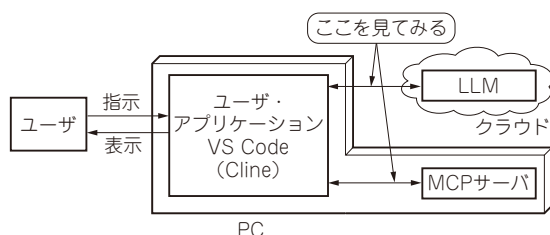


図1 ユーザ・アプリケーションがどのようにLLMやMCPサーバと通信しているのか実際に見てみる

● Hello Worldのプログラムを作る

図2(a)は文字を出力するプログラムを作成した際の画面です。下側チャット欄に「pythonでhello worldを出力するプログラムを作成して」と入力します。[Enter]キーで、図2(b)の画面に変わります。API Requestの部分はモデルからの応答文で、要件確認や手法検討を行っています。右側には、hello.pyファイルをprint("HelloWorld")という内容で作成しようとしている様子が表示されています。



(a) チャット欄にプロンプトを入力する画面



(b) モデルからの応答文の表示画面

図2 Clineの対話画面

ClineでPythonのhello worldプログラムを作成したときの様子