Model Context Protocol: An In-Depth Technical Report

1. Introduction to Model Context Protocol (MCP)

The integration of Artificial Intelligence (AI) applications with external systems has traditionally presented a complex challenge. The Model Context Protocol (MCP) emerges as an open standard, spearheaded by Anthropic, with the explicit aim of streamlining and standardizing the manner in which AI applications, encompassing chatbots, Integrated Development Environment (IDE) assistants, and bespoke agents, establish connections with a diverse array of external tools, data repositories, and systems.[1] This protocol seeks to address the inherent complexities arising from the integration of multiple AI applications with numerous external resources, a scenario often characterized as the "M×N problem," wherein M represents the number of AI applications and N the number of tools or systems. The conventional approach to this problem often necessitates the development of M×N distinct integrations, leading to substantial redundancy in effort and inconsistencies in implementation.[2]

MCP offers a transformative solution by providing a common Application Programming Interface (API) that reframes the integration challenge into an "M+N problem." This paradigm shift requires tool developers to construct N MCP servers, one for each system they wish to expose, while application developers build M MCP clients, one for each AI application they are developing.[2] This standardized approach facilitates a more efficient and scalable ecosystem for AI integrations, akin to the universal connectivity provided by the Universal Serial Bus (USB) standard for computer peripherals.[3] Before the advent of USB, connecting various peripherals to a computer necessitated a multitude of different ports and bespoke drivers. Similarly, the integration of AI applications with external functionalities has historically been a fragmented landscape of custom solutions.[2] MCP endeavors to provide a unified and plug-and-play mechanism for AI to interact with a wide spectrum of data sources and tools, mirroring the simplicity and versatility of USB-C connectivity.[7]

The adoption of MCP holds the promise of accelerating the development of sophisticated AI agents and intricate workflows built upon the foundation of Large Language Models (LLMs). By offering a standardized protocol, MCP fosters the creation of a burgeoning ecosystem of pre-built integrations, thereby enhancing the flexibility and adaptability of AI applications across different LLM providers and vendors.[3] Notably, early adopters and companies specializing in development tools are actively integrating MCP into their platforms. This integration empowers AI agents with enhanced capabilities for retrieving pertinent information and achieving a deeper understanding of contextual nuances, ultimately leading to more relevant and accurate responses.[4] Furthermore, MCP facilitates the ability of AI assistants to execute tangible actions within various platforms by establishing direct connections with external data sources, enabling both the reading and writing of data.[10] The protocol also introduces a structured methodology for LLMs to maintain, update, and access contextual information, thereby enabling autonomous management of workflows and significantly improving the operational efficiency of these models.[10]

2. Core Concepts of MCP

The Model Context Protocol operates on a client-server architectural model, wherein host applications, such as Claude Desktop or various IDEs, incorporate MCP clients that can establish connections with multiple MCP servers.[3] Within this architecture, each client maintains a dedicated, one-to-one connection with a specific MCP server.[2] The servers themselves are external programs designed to expose functionalities, data, and interaction templates to the AI model via the intermediary client.[2] This design fosters a clear separation of concerns, allowing LLM applications to concentrate on their core reasoning and user interaction functionalities while offloading tasks related to external data retrieval, tool execution, and specialized computations to dedicated server components.[13]

Within the MCP framework, distinct roles are assigned to different components. **Hosts** are the primary applications through which users interact with AI, examples including Claude Desktop, IDEs like Cursor, and custom-built agents.[2] These hosts are responsible for coordinating the overall system and managing the interactions of the underlying LLMs.[7] **Clients** reside within the host applications and are tasked with managing the connection to a specific MCP server. Their responsibilities encompass connection establishment, discovery of server capabilities, forwarding requests from the host to the server, and handling responses from the server back to the host.[2] In essence, clients act as the communication bridge between the AI within the host and the external capabilities offered by the servers.[14] **Servers**, on the other hand, are standalone programs that expose specialized capabilities through a standardized API defined by the MCP. They serve as the crucial link between the AI world of the protocol and the specific functionalities of external systems, whether those are APIs, databases, or local file systems.[2] These servers provide a range of features accessible via the MCP, including tools, resources, and prompts.[7]

The functionalities exposed by MCP servers are categorized into three core concepts: **Tools**, **Resources**, and **Prompts**.[2] **Tools** are model-controlled functions that an LLM can invoke to perform specific actions, such as querying a weather API. This is analogous to the concept of function calling in other AI frameworks.[2] **Resources** are application-controlled data sources that an LLM can access, similar to making GET requests to a REST API. These resources provide data without significant computational overhead or side effects and are typically used to supply context to the LLM's request.[2] **Prompts** are user-controlled, pre-defined templates designed to optimize the use of tools or resources. Users select these prompts before initiating the inference process, guiding the LLM in how to best interact with the available capabilities.[2]

The lifecycle of an MCP connection begins with an **Initialization** phase. When a host application starts and instantiates MCP clients, these clients initiate a handshake process with their respective servers to exchange information about their capabilities and the protocol versions they support.[2] This involves the client sending an initialize request to the server, which includes the client's supported protocol version and its own capabilities. The server then responds with its supported protocol version and the capabilities it offers. Finally, the client sends an initialized notification to the server to acknowledge the successful completion of the initialization process.[11] Following initialization, the **Discovery** phase allows clients to ascertain the specific functionalities offered by the connected servers. Clients can send requests to the server to list the available tools, resources, and prompts, along with their descriptions.[2] For instance, clients can use the tools/list endpoint to retrieve a catalog of executable tools provided by the server.[13] Once the client has discovered the server's capabilities and the LLM within the host has determined the need to utilize an external function or data source, the **Completion** phase occurs. The client relays the results of tool execution or resource access back to the host application. The host then integrates this information into the LLM's context, enabling the LLM to generate a final, informed response for the user.[2]

4. Detailed Explanation of MCP Module Contents

- **mcp.client Module Functions:**
  - CallToolRequest(): This function, or its class counterpart, is invoked by the client to execute a specific tool provided by the server.[16] It encapsulates the name of the tool to be called and any arguments required for its execution. For example, to instruct a server to fetch the current weather, a client might use CallToolRequest() with the tool name "get_weather" and arguments specifying the location.
  - ClientCapabilities(): This function or class defines the capabilities that an MCP client supports.[4] These capabilities, such as support for file system roots or LLM sampling, are advertised to the server during the initialization phase, allowing the server to tailor its interactions based on the client's abilities.

- ClientNotification(): This serves as a base class for various notifications initiated by the client and sent to the server.[33] These are asynchronous messages that do not require a response. A specific example is the InitializedNotification, which the client sends to the server upon completing its initialization process.
- ClientRequest(): This is a base class for requests originating from the client that expect a response from the server.[37] Examples of concrete requests that would inherit from this base include ListToolsRequest (to get a list of available tools) and ReadResourceRequest (to read the content of a specific resource).
- ClientResult(): This base class defines the structure for results sent by the server in response to client requests.[36] Concrete results such as ListToolsResult (containing the list of tools) and ReadResourceResult (containing the resource content) would be based on this class.
- ClientSession(): This function or class is responsible for managing the entire session lifecycle between an MCP client and a server.[2] It handles connection establishment, message routing, and session-specific state.
- CompleteRequest(): This function or class represents a request from the client to the server to retrieve options for autocompletion.[23] This is useful in scenarios where a user is interacting with a prompt or tool argument and the client needs suggestions from the server.
- CreateMessageRequest(): This function or class defines a request initiated by the server to the client, asking the client to use its LLM to sample a message.[17] This is a key part of the sampling capability, allowing servers to leverage client-side LLM functionalities.
- CreateMessageResult(): This function or class encapsulates the client's response to a CreateMessageRequest from the server.[17] It contains the message sampled by the client's LLM, which is then returned to the server.
- ErrorData(): This function or class is used to structure the error information included in JSON-RPC error responses.[28] It typically includes an error code, a message, and optionally, additional data providing context about the error.
- GetPromptRequest(): This function or class represents a request from the client to the server to retrieve a specific prompt template.[17] The client specifies the name of the desired prompt and can also include arguments to be used with the template.
- GetPromptResult(): This function or class defines the server's response to a GetPromptRequest from the client.[17] It typically contains a description of the prompt and a list of messages that constitute the prompt.
- Implementation(): This function or class is used to describe the specific implementation of an MCP client or server, including its name and version.[3]

This information is exchanged during the initialization handshake.

- IncludeContext(): This function or class is likely used within the parameters of a sampling request to specify which context from the MCP environment should be included when the client's LLM generates a message.[13] Options might include no context, context from the current server, or context from all connected servers.
- InitializeRequest(): This function or class represents the initial request sent by the client to the server upon establishing a connection.[11] It includes the client's protocol version and its capabilities.
- InitializeResult(): This function or class defines the response sent by the server to the client's InitializeRequest.[16] It contains the server's protocol version and its supported capabilities.
- InitializedNotification(): This function or class represents the notification sent by the client to the server after the client has finished its initialization process.[7] It signals to the server that the client is ready for normal operation.
- JSONRPCError(): This function or class is used to construct a JSON-RPC error response to a request.[28] It includes fields for the error code, message, and optional data.
- JSONRPCRequest(): This function or class defines the structure of a JSON-RPC request, which expects a corresponding response.[9] It includes fields for the JSON-RPC version, request ID, method name, and parameters.
- JSONRPCResponse(): This function or class defines the structure of a successful JSON-RPC response to a request.[35] It includes fields for the JSON-RPC version, request ID, and the result.
- ListPromptsRequest(): This function or class represents a request from the client to the server to get a list of available prompts and prompt templates.[23] It may include parameters for pagination.
- ListPromptsResult(): This function or class defines the server's response to a ListPromptsRequest, containing a list of prompts and an optional cursor for pagination.[17]
- ListResourcesRequest(): This function or class represents a request from the client to the server to get a list of available resources.[23] It may include parameters for pagination.
- ListResourcesResult(): This function or class defines the server's response to a ListResourcesRequest, containing a list of resources with their metadata and an optional cursor for pagination.[17]
- ListToolsResult(): This function or class defines the server's response to a tools/list request, containing a list of available tools with their specifications and an optional cursor for pagination.[17]

- LoggingLevel(): This function or class likely defines an enumeration or set of constants representing the different logging levels supported by MCP, such as DEBUG, INFO, WARNING, ERROR, etc..[17]
- LoggingMessageNotification(): This function or class is used to construct a notification message containing log information sent from the server to the client.[18] It includes the log level, logger name, and the log message data.
- McpError(): This function or class likely defines a custom exception type that is raised when an error occurs during MCP communication.[156]
- Notification(): This is a base class for all notification messages in MCP, following the JSON-RPC 2.0 standard.[11]
- PingRequest(): This function or class represents a request sent to check the liveness of the other party in the MCP connection (client or server).[34]
- ProgressNotification(): This function or class is used to construct an out-of-band notification to inform the receiver about the progress of a long-running operation.[34] It includes a progress token, current progress, and optional total and message.
- PromptsCapability(): This function or class likely defines a structure representing the prompts capability, which a server can advertise during initialization to indicate its support for prompt-related operations.[35]
- ReadResourceRequest(): This function or class represents a request from the client to the server to read the content of a specific resource identified by its URI.[17]
- ReadResourceResult(): This function or class defines the server's response to a ReadResourceRequest, containing the content of the requested resource, which can be text or binary data.[17]
- Resource(): This function or class likely defines a structure representing a resource exposed by the server, including its URI, name, and other metadata.[2]
- ResourceUpdatedNotification(): This function or class is used to construct a notification sent by the server to the client to inform it that a specific resource has been updated.[34]
- ResourcesCapability(): This function or class likely defines a structure representing the resources capability, which a server can advertise to indicate its support for resource-related operations, such as listing and reading resources.[35]
- RootsCapability(): This function or class likely defines a structure representing the roots capability, which a client can advertise to the server to indicate its support for managing file system roots.[18]
- SamplingMessage(): This function or class defines the structure of a message exchanged during the LLM sampling process.[35] It typically includes the role

(user or assistant) and the content of the message.

- SamplingRole(): This function or class likely defines an enumeration or set of constants representing the roles of messages in the sampling interaction, such as 'user' and 'assistant'.[101]
- ServerCapabilities(): This function or class defines the capabilities that an MCP server supports.[61] These are advertised to the client during the initialization phase.
- ServerNotification(): This serves as a base class for various notifications initiated by the server and sent to the client.[37] Examples include LoggingMessageNotification and ResourceUpdatedNotification.
- ServerRequest(): This is a base class for requests originating from the server that expect a response from the client.[35] An example is ListRootsRequest.
- ServerResult(): This base class defines the structure for results sent by the client in response to server requests.[35] An example is ListRootsResult.
- ServerSession(): This function or class is responsible for managing the session lifecycle on the server side.[14] It handles connection management and session-specific data for each client.
- SetLevelRequest(): This function or class represents a request from the client to the server to set the logging level.[23] The client specifies the desired logging level.
- StdioServerParameters(): This function or class is used to configure the parameters for an MCP server that communicates with a client over standard input and output streams.[2] This includes the command to execute to start the server and any command-line arguments.
- StopReason(): This function or class likely defines an enumeration or set of constants representing the reasons why an LLM might stop generating a response during sampling, such as reaching the end of turn, encountering a stop sequence, or hitting the maximum token limit.[37]
- SubscribeRequest(): This function or class represents a request from the client to the server to subscribe to updates for a specific resource.[34] The client specifies the URI of the resource it wants to subscribe to.
- Tool(): This function or class defines the structure for a tool exposed by an MCP server.[11] It includes the tool's name, a human-readable description, and a JSON schema defining the expected input parameters.
- ToolsCapability(): This function or class likely defines a structure representing the tools capability, which a server can advertise to indicate its support for tool-related operations, such as listing and calling tools.[35]
- UnsubscribeRequest(): This function or class represents a request from the client to the server to cancel a previous subscription to resource updates.[7]

The client specifies the URI of the resource from which it wants to unsubscribe.

- **MCP Module Attributes:**
  - client: This attribute refers to a module within the MCP library that contains the implementation for the client side of the protocol.
  - server: This attribute refers to a module within the MCP library that contains the implementation for the server side of the protocol.
  - shared: This attribute refers to a module that contains code and data structures shared between the client and server implementations.
  - types: This attribute refers to a module that likely defines all the custom data types and message schemas used throughout the MCP protocol.
  - stdio_client: This attribute likely provides a specific implementation of an MCP client that uses the standard input/output streams for communication with a server, suitable for local processes.
  - stdio_server: This attribute likely provides a specific implementation of an MCP server that uses the standard input/output streams for communication with a client, also suitable for local processes.

5. Relationships Between MCP Components

The Model Context Protocol facilitates interaction between AI applications (clients) and external systems (servers) through a structured exchange of messages. A fundamental relationship is the **request-response flow**.[11] The MCP client, residing within a host application, initiates communication by sending a ClientRequest, such as a request to list available tools (ListToolsRequest) or to read a specific resource (ReadResourceRequest). This request is transmitted to the designated MCP server. Upon receiving the request, the server processes it based on its advertised capabilities and the specific handlers implemented for the requested method. Following the processing, the server sends back a ServerResult, which could be a ListToolsResult containing the list of tools or a ReadResourceResult containing the content of the resource. In cases where an error occurs during the server's processing of the request, it communicates this back to the client using a standardized JSONRPCError response.[11] This mechanism ensures that the client is informed about the outcome of its requests, whether successful or not.

Another crucial relationship lies in the **capability exchange** that occurs during the initialization phase of an MCP connection.[11] When a client first connects to a server, it sends its ClientCapabilities, detailing the features it supports, such as the ability to handle file system roots (roots) or to perform LLM sampling on behalf of the server (sampling).[11] The server, in turn, responds with its ServerCapabilities, indicating the types of functionalities it can offer, including the availability of tools, resources, and prompts.[11] This exchange of capability information is vital as it informs the server about the client's limitations and supported features, thereby guiding the server's behavior in subsequent interactions.[17] For instance, if a client does not declare support for the roots capability, the server will refrain from attempting to utilize any root-related functionalities during the session.

Beyond the request-response paradigm, MCP also supports asynchronous communication through **notifications**.[11] Servers can proactively send ServerNotification messages to clients to provide updates on events that have occurred, such as a LoggingMessageNotification conveying log information or a ResourceUpdatedNotification indicating that the content of a subscribed resource has changed.[11] Similarly, clients can send ClientNotification messages to the server, for example, the InitializedNotification sent after the client has completed its setup or a RootsListChangedNotification to inform the server about changes in the client's accessible file system roots.[11] This asynchronous communication allows for a more dynamic and event-driven interaction between MCP components, enabling real-time updates and background processes without the overhead of explicit requests and responses.

6. Practical Scenario: Tool Invocation

Consider a scenario where a user, working within an IDE that supports MCP (the host application), wants to leverage a specific tool provided by an external MCP server. This tool might be designed to interact with a version control system, such as Git, allowing the user to perform actions like committing code changes directly through the IDE. The sequence of MCP messages involved in this interaction would typically unfold as follows:

1. **Client Initialization:** Upon starting the IDE, the integrated MCP client establishes a connection with the Git MCP server.[2] The client initiates the handshake by sending an InitializeRequest to the server, advertising its protocol version and capabilities, which would include support for tools, resources, and prompts.
2. **Server Initialization Response:** The Git MCP server receives the InitializeRequest and responds with an InitializeResult.[11] This response confirms the protocol version and indicates the server's capabilities, notably that it offers tools for interacting with Git repositories.
3. **Client Sends Initialized Notification:** Following the successful receipt and processing of the InitializeResult, the client sends an InitializedNotification to the server.[11] This notification signifies that the client is ready to engage in normal

communication.

4. **Client Requests List of Tools:** To discover the specific Git-related actions it can perform, the client sends a ListToolsRequest to the server.[12] This request asks the server to provide a catalog of the tools it makes available.
5. **Server Responds with List of Tools:** The Git MCP server processes the ListToolsRequest and returns a ListToolsResult.[12] This message contains a list of tools, which might include tools for committing changes (commit), creating branches (create_branch), viewing repository status (status), and so forth, along with their descriptions and the parameters they accept.
6. **LLM Determines to Use a Tool:** The user, working in the IDE, makes changes to their code and then instructs the IDE's AI agent to commit these changes with a specific message. The LLM, aware of the available Git tools from the ListToolsResult, determines that the commit tool is the most appropriate action.[2]
7. **Client Sends CallToolRequest:** The IDE's MCP client, acting on the LLM's decision, sends a CallToolRequest to the Git MCP server.[12] This request specifies the name of the tool as commit and includes the commit message as an argument.
8. **Server Executes Tool and Sends Result:** The Git MCP server receives the CallToolRequest, executes the commit tool using the provided commit message, and then sends back a CallToolResult.[12] This result indicates whether the commit was successful or if any errors occurred.
9. **Client Relays Result to Host:** The IDE's MCP client receives the CallToolResult from the Git MCP server and relays this information back to the IDE and the LLM.[2]
10. **Host Provides Response to User:** Finally, the IDE, based on the CallToolResult, provides feedback to the user, such as a confirmation message if the commit was successful or an error notification if it failed.[2]

7. Conclusion

The Model Context Protocol represents a significant stride towards achieving seamless and standardized integration between AI applications and the vast ecosystem of external tools and data sources. By establishing a clear architectural framework based on client-server communication and defining a comprehensive set of message types and capabilities, MCP effectively addresses the complexities inherent in AI integrations. The protocol's emphasis on dynamic discovery, capability negotiation, and asynchronous notifications allows for flexible and adaptable interactions between diverse AI applications and a wide range of specialized servers. The practical scenario of tool invocation underscores the structured and efficient communication facilitated by MCP, highlighting its potential to enhance the functionality and user experience of AI-powered tools across various domains. As the MCP ecosystem continues to evolve, with growing adoption by major AI players and an expanding repository of community-driven servers, it is poised to become a cornerstone for building robust, secure, and highly modular AI applications in the future.

Works cited

1. www.philschmid.de, accessed on April 15, 2025, https://www.philschmid.de/mcp-introduction#:~:text=The%20Model%20Context%20Protocol%20(MCP,like%20USB%20for%20AI%20integrations.
2. Model Context Protocol (MCP) an overview - Philschmid, accessed on April 15, 2025, https://www.philschmid.de/mcp-introduction
3. Model Context Protocol: Introduction, accessed on April 15, 2025, https://modelcontextprotocol.io/introduction
4. Introducing the Model Context Protocol - Anthropic, accessed on April 15, 2025, https://www.anthropic.com/news/model-context-protocol
5. Model Context Protocol (MCP) - Anthropic API, accessed on April 15, 2025, https://docs.anthropic.com/en/docs/agents-and-tools/mcp
6. An Introduction to Model Context Protocol - MCP 101 - DigitalOcean, accessed on April 15, 2025, https://www.digitalocean.com/community/tutorials/model-context-protocol
7. What is Model Context Protocol (MCP): Explained - Composio, accessed on April 15, 2025, https://composio.dev/blog/what-is-model-context-protocol-mcp-explained/
8. What Is the Model Context Protocol (MCP) and How It Works - Descope, accessed on April 15, 2025, https://www.descope.com/learn/post/mcp
9. Model Context Protocol (MCP): A comprehensive introduction for developers - Stytch, accessed on April 15, 2025, https://stytch.com/blog/model-context-protocol-introduction/
10. What you need to know about the Model Context Protocol (MCP) - Merge, accessed on April 15, 2025, https://www.merge.dev/blog/model-context-protocol
11. Core architecture - Model Context Protocol, accessed on April 15, 2025, https://modelcontextprotocol.io/docs/concepts/architecture
12. model-context-protocol-resources/guides/mcp-client-development-guide.md at main, accessed on April 15, 2025, https://github.com/cyanheads/model-context-protocol-resources/blob/main/guides/mcp-client-development-guide.md
13. model-context-protocol-resources/guides/mcp-server-development-guide.md at main - GitHub, accessed on April 15, 2025, https://github.com/cyanheads/model-context-protocol-resources/blob/main/guides/mcp-server-development-guide.md
14. Model Context Protocol (MCP): Integrating Azure OpenAI for Enhanced Tool Integration and Prompting - Microsoft Tech Community, accessed on April 15,

2025,
https://techcommunity.microsoft.com/blog/azure-ai-services-blog/model-context-protocol-mcp-integrating-azure-openai-for-enhanced-tool-integratio/4393788
15. Everything a Developer Needs to Know About the Model Context Protocol (MCP) - Neo4j, accessed on April 15, 2025, https://neo4j.com/blog/developer/model-context-protocol/
16. mcp-nutanix - Glama, accessed on April 15, 2025, https://glama.ai/mcp/servers/@thunderboltsid/mcp-nutanix/blob/main/main.go
17. MCP Client - Model Context Protocol, accessed on April 15, 2025, https://modelcontextprotocol.io/sdk/java/mcp-client
18. Java MCP SDK :: Spring AI Reference, accessed on April 15, 2025, https://docs.spring.io/spring-ai-mcp/reference/mcp.html
19. MCP Part III - Application: Custom Server for a Specific Use Case - Olivier Wulveryck, accessed on April 15, 2025, https://blog.owulveryck.info/2025/01/27/mcp-part-iii-application-custom-server-for-a-specific-use-case.html
20. Building a Generative AI MCP Client Application in Go using Ollama, accessed on April 15, 2025, https://k33g.hashnode.dev/building-a-generative-ai-mcp-client-application-in-go-using-ollama
21. mark3labs/mcp-go: A Go implementation of the Model Context Protocol (MCP), enabling seamless integration between LLM applications and external data sources and tools. - GitHub, accessed on April 15, 2025, https://github.com/mark3labs/mcp-go
22. mcp - Go Packages, accessed on April 15, 2025, https://pkg.go.dev/github.com/sunfmin/mcp-go-debugger/pkg/mcp
23. client package - github.com/mark3labs/mcp-go/client - Go Packages, accessed on April 15, 2025, https://pkg.go.dev/github.com/mark3labs/mcp-go/client
24. mcp-grafana/tools_test.go at main - GitHub, accessed on April 15, 2025, https://github.com/grafana/mcp-grafana/blob/main/tools_test.go
25. CallToolRequest in mcp_core::tools - Rust - Docs.rs, accessed on April 15, 2025, https://docs.rs/mcp-core/latest/mcp_core/tools/struct.CallToolRequest.html
26. ClientCapabilities in mcp_attr::schema - Rust - Docs.rs, accessed on April 15, 2025, https://docs.rs/mcp-attr/latest/mcp_attr/schema/struct.ClientCapabilities.html
27. Base MCP Server | Glama, accessed on April 15, 2025, https://glama.ai/mcp/servers/@base/base-mcp
28. Mcp client does not support `roots/list` - Bug Reports - Cursor - Community Forum, accessed on April 15, 2025, https://forum.cursor.com/t/mcp-client-does-not-support-roots-list/77248
29. punkpeye/fastmcp: A TypeScript framework for building MCP servers. - GitHub, accessed on April 15, 2025, https://github.com/punkpeye/fastmcp
30. [Client] Implement proper roots and sampling client capabilities abstraction · Issue #13 · cloudwalk/hermes-mcp - GitHub, accessed on April 15, 2025, https://github.com/cloudwalk/hermes-mcp/issues/13

31. MCP Unrecognized field \"tools\ - Bug Reports - Cursor - Community Forum, accessed on April 15, 2025, https://forum.cursor.com/t/mcp-unrecognized-field-tools/65513
32. client package - github.com/WePrompt/gomcp/client - Go Packages, accessed on April 15, 2025, https://pkg.go.dev/github.com/WePrompt/gomcp/client
33. ClientNotification in mcp_attr::schema - Rust - Docs.rs, accessed on April 15, 2025, https://docs.rs/mcp-attr/latest/mcp_attr/schema/enum.ClientNotification.html
34. mcp-openapi-server/mcp-specs.d.ts at main - GitHub, accessed on April 15, 2025, https://github.com/ivo-toby/mcp-openapi-server/blob/main/mcp-specs.d.ts
35. mcp_schema - Rust - Docs.rs, accessed on April 15, 2025, https://docs.rs/mcp-schema
36. MCP Kotlin SDK, accessed on April 15, 2025, https://jetbrains.github.io/mcp-kotlin-sdk/
37. mcp - Go Packages, accessed on April 15, 2025, https://pkg.go.dev/github.com/mark3labs/mcp-go/mcp
38. protocol package - github.com/ThinkInAIXYZ/go-mcp/protocol - Go Packages, accessed on April 15, 2025, https://pkg.go.dev/github.com/ThinkInAIXYZ/go-mcp/protocol
39. python-sdk/src/mcp/client/session.py at main - GitHub, accessed on April 15, 2025, https://github.com/modelcontextprotocol/python-sdk/blob/main/src/mcp/client/session.py
40. Memorandum of Understanding (MOU) with Inland Empire Health Plan (IEHP), a Managed Care Plan (MCP), is intended, accessed on April 15, 2025, https://www.iehp.org/content/dam/iehp-org/en/documents/joint-agency-meetings/quarter-1/san-bernardino/SBDBH%20MOU%20And%20Eating%20Disorder%20Addendum%20Combined.pdf
41. PostgreSQL MCP Server - Glama, accessed on April 15, 2025, https://glama.ai/mcp/servers/@hsinyuyen/my-postgres-mcp/tree/main/node_modules
42. 500 server error for MCP - Archon - oTTomator Community, accessed on April 15, 2025, https://thinktank.ottomator.ai/t/500-server-error-for-mcp/6697
43. In-Depth review of the MCP authorization spec (2025-03-26 edition) - Logto blog, accessed on April 15, 2025, https://blog.logto.io/mcp-auth-spec-review-2025-03-26
44. McpClientAutoConfiguration (Spring AI 1.0.0-M6 API), accessed on April 15, 2025, https://docs.spring.io/spring-ai/docs/1.0.0-M6/api/org/springframework/ai/autoconfigure/mcp/client/McpClientAutoConfiguration.html
45. rust_mcp_schema - Rust - Docs.rs, accessed on April 15, 2025, https://docs.rs/rust-mcp-schema
46. Model Context Protocol (MCP) Hands-On Walkthrough for the Unstructured Workflow Endpoint, accessed on April 15, 2025, https://docs.unstructured.io/examplecode/tools/mcp
47. MCP Transport Mechanisms - Cline Documentation, accessed on April 15, 2025,

https://docs.cline.bot/mcp-servers/mcp-transport-mechanisms
48. Model Context Protocol (MCP) Quickstart - Glama, accessed on April 15, 2025, https://glama.ai/blog/2024-11-25-model-context-protocol-quickstart
49. Pass entire request object to handlers; add raw request to MCP base Request · Issue #195 · modelcontextprotocol/python-sdk - GitHub, accessed on April 15, 2025, https://github.com/modelcontextprotocol/python-sdk/issues/195
50. ClientResult in mcp_attr::schema - Rust - Docs.rs, accessed on April 15, 2025, https://docs.rs/mcp-attr/latest/mcp_attr/schema/enum.ClientResult.html
51. MCP OpenAI Server | Glama, accessed on April 15, 2025, https://glama.ai/mcp/servers/riynwmasx9/tree/main/docs
52. jlowin/fastmcp: The fast, Pythonic way to build MCP servers and clients - GitHub, accessed on April 15, 2025, https://github.com/jlowin/fastmcp
53. Example Clients - Model Context Protocol, accessed on April 15, 2025, https://modelcontextprotocol.io/clients
54. With all the MCP servers over 2000 now and counting, which are the MCP clients people are using - Reddit, accessed on April 15, 2025, https://www.reddit.com/r/mcp/comments/1jde28x/with_all_the_mcp_servers_over_2000_now_and/
55. A Developer's Guide to the MCP - Zep - AI, accessed on April 15, 2025, https://www.getzep.com/ai-agents/developer-guide-to-mcp?ref=blog.getzep.com
56. Mcp - LlamaIndex, accessed on April 15, 2025, https://docs.llamaindex.ai/en/stable/api_reference/tools/mcp/
57. For Client Developers - Model Context Protocol, accessed on April 15, 2025, https://modelcontextprotocol.io/quickstart/client
58. MCP Servers - Chainlit, accessed on April 15, 2025, https://docs.chainlit.io/advanced-features/mcp
59. Build and deploy Remote Model Context Protocol (MCP) servers to Cloudflare, accessed on April 15, 2025, https://blog.cloudflare.com/remote-model-context-protocol-servers-mcp/
60. mcp-http-client-example/main.py at main - GitHub, accessed on April 15, 2025, https://github.com/slavashvets/mcp-http-clilent-example/blob/main/main.py
61. The official Python SDK for Model Context Protocol servers and clients - GitHub, accessed on April 15, 2025, https://github.com/modelcontextprotocol/python-sdk
62. CompleteRequest in mcp_attr::schema - Rust - Docs.rs, accessed on April 15, 2025, https://docs.rs/mcp-attr/latest/mcp_attr/schema/struct.CompleteRequest.html
63. Completion - Model Context Protocol Specification, accessed on April 15, 2025, https://spec.modelcontextprotocol.io/specification/2024-11-05/server/utilities/completion/
64. [Server] Implement completion feature · Issue #15 · cloudwalk/hermes-mcp - GitHub, accessed on April 15, 2025, https://github.com/cloudwalk/hermes-mcp/issues/15
65. Sampling - Model Context Protocol Specification, accessed on April 15, 2025, https://spec.modelcontextprotocol.io/specification/2024-11-05/client/sampling/

66. HttpApplication.CompleteRequest Method (System.Web) | Microsoft Learn, accessed on April 15, 2025, https://learn.microsoft.com/en-us/dotnet/api/system.web.httpapplication.completerequest?view=netframework-4.8.1
67. mcp-figma - Glama, accessed on April 15, 2025, https://glama.ai/mcp/servers/z3gvv0394m/tree/main/node_modules/@modelcontextprotocol
68. Microsoft partners with Anthropic to create official C# SDK for Model Context Protocol, accessed on April 15, 2025, https://devblogs.microsoft.com/blog/microsoft-partners-with-anthropic-to-create-official-c-sdk-for-model-context-protocol
69. CreateMessageRequest class - mcp_server library - Dart API - Pub.dev, accessed on April 15, 2025, https://pub.dev/documentation/mcp_server/latest/mcp_server/CreateMessageRequest-class.html
70. MCP Client Boot Starter :: Spring AI Reference, accessed on April 15, 2025, https://docs.spring.io/spring-ai/reference/api/mcp/mcp-client-boot-starter-docs.html
71. MCP Server - Model Context Protocol, accessed on April 15, 2025, https://modelcontextprotocol.io/sdk/java/mcp-server
72. sovran-mcp - crates.io: Rust Package Registry, accessed on April 15, 2025, https://crates.io/crates/sovran-mcp
73. java-sdk/migration-0.8.0.md at main - GitHub, accessed on April 15, 2025, https://github.com/modelcontextprotocol/java-sdk/blob/main/migration-0.8.0.md
74. McpAsyncClient.java - GitHub, accessed on April 15, 2025, https://github.com/spring-projects-experimental/spring-ai-mcp/blob/main/mcp/src/main/java/org/springframework/ai/mcp/client/McpAsyncClient.java
75. MCP Client Boot Starter :: Spring AI Reference, accessed on April 15, 2025, https://docs.spring.io/spring-ai/reference/1.0/api/mcp/mcp-client-boot-starter-docs.html
76. mcp-agent/src/mcp_agent/workflows/llm/augmented_llm.py at main - GitHub, accessed on April 15, 2025, https://github.com/lastmile-ai/mcp-agent/blob/main/src/mcp_agent/workflows/llm/augmented_llm.py
77. README.md - MCP Python SDK - GitHub, accessed on April 15, 2025, https://github.com/modelcontextprotocol/python-sdk/blob/main/README.md
78. types package - github.com/dwrtz/mcp-go/pkg/types - Go Packages, accessed on April 15, 2025, https://pkg.go.dev/github.com/dwrtz/mcp-go/pkg/types
79. Sentry MCP Server - Glama, accessed on April 15, 2025, https://glama.ai/mcp/servers/sfel2251tz
80. What the heck is MCP? And why is everybody talking about it? - Pieces for developers, accessed on April 15, 2025, https://pieces.app/blog/mcp?__hstc=81555528.2f3f33a24b44870ec4a577029c49e44b.1733616000144.1733616000145.1733616000146.1&__hssc=81555528.1.1733616000147&__hsfp=2637229211

81. Building Custom Extensions with Goose - GitHub Pages, accessed on April 15, 2025, https://block.github.io/goose/docs/tutorials/custom-extensions/
82. Sentry MCP Server – A Model Context Protocol server that enables AI assistants to interact with Sentry for error tracking and monitoring, allowing retrieval and analysis of error data, project management, and performance monitoring through the Sentry API. - Reddit, accessed on April 15, 2025, https://www.reddit.com/r/mcp/comments/1j6w446/sentry_mcp_server_a_model_context_protocol_server/
83. AI Model Context Protocol (MCP) and Security - Cisco Community, accessed on April 15, 2025, https://community.cisco.com/t5/security-blogs/ai-model-context-protocol-mcp-and-security/ba-p/5274394
84. JigsawStack MCP Servers: Bridging LLMs with Context and Tools, accessed on April 15, 2025, https://jigsawstack.com/blog/jigsawstack-mcp-servers
85. servers/src/fetch/src/mcp_server_fetch/server.py at main - GitHub, accessed on April 15, 2025, https://github.com/modelcontextprotocol/servers/blob/main/src/fetch/src/mcp_server_fetch/server.py
86. Payman AI MCP Server - Glama, accessed on April 15, 2025, https://glama.ai/mcp/servers/@hrishi0102/payman_mcp/blob/main/payman-server.ts
87. GetPromptRequest in mcp_attr::schema - Rust - Docs.rs, accessed on April 15, 2025, https://docs.rs/mcp-attr/latest/mcp_attr/schema/struct.GetPromptRequest.html
88. MCP Server Boot Starter :: Spring AI Reference, accessed on April 15, 2025, https://docs.spring.io/spring-ai/reference/1.0/api/mcp/mcp-server-boot-starter-docs.html
89. mcp-go/examples/everything/main.go at main - GitHub, accessed on April 15, 2025, https://github.com/mark3labs/mcp-go/blob/main/examples/everything/main.go
90. GetPromptResult in mcp_attr::schema - Rust - Docs.rs, accessed on April 15, 2025, https://docs.rs/mcp-attr/latest/mcp_attr/schema/struct.GetPromptResult.html
91. McpGetPromptResult (LangChain4j :: Aggregator 1.0.0-beta3 API), accessed on April 15, 2025, https://docs.langchain4j.dev/apidocs/dev/langchain4j/mcp/client/McpGetPromptResult.html
92. Prompts - Model Context Protocol, accessed on April 15, 2025, https://modelcontextprotocol.io/docs/concepts/prompts
93. MCP Server Boot Starter :: Spring AI Reference, accessed on April 15, 2025, https://docs.spring.io/spring-ai/reference/api/mcp/mcp-server-boot-starter-docs.html
94. Model Context Protocol (MCP) :: Spring AI Reference, accessed on April 15, 2025, https://docs.spring.io/spring-ai/reference/api/mcp/mcp-overview.html
95. Overview - Model Context Protocol, accessed on April 15, 2025,

https://modelcontextprotocol.io/sdk/java/mcp-overview
96. What Is the Model Context Protocol (MCP) and How It Works - Descope, accessed on April 15, 2025, https://www.descope.com/learn/post/mcp?ref=jeffreybowdoin.com
97. What is MCP (Model Context Protocol) and how it works - Logto blog, accessed on April 15, 2025, https://blog.logto.io/what-is-mcp
98. Model Context Protocol (MCP) - LangChain4j, accessed on April 15, 2025, https://docs.langchain4j.dev/tutorials/mcp/
99. Model Context Protocol - GitHub, accessed on April 15, 2025, https://github.com/modelcontextprotocol
100. modelcontextprotocol/servers: Model Context Protocol Servers - GitHub, accessed on April 15, 2025, https://github.com/modelcontextprotocol/servers
101. Sampling - Model Context Protocol, accessed on April 15, 2025, https://modelcontextprotocol.io/docs/concepts/sampling
102. Model Context Protocol (MCP): A Guide for QA Teams | TestCollab Blog, accessed on April 15, 2025, https://testcollab.com/blog/model-context-protocol-mcp-a-guide-for-qa-teams
103. A Journey from AI to LLMs and MCP - 10 - Sampling and Prompts in MCP — Making Agent Workflows Smarter and Safer - DEV Community, accessed on April 15, 2025, https://dev.to/alexmercedcoder/a-journey-from-ai-to-llms-and-mcp-10-sampling-and-prompts-in-mcp-making-agent-workflows-2446
104. CreateMessageRequestParams class - mcp_dart library - Dart API - Pub.dev, accessed on April 15, 2025, https://pub.dev/documentation/mcp_dart/latest/mcp_dart/CreateMessageRequestParams-class.html
105. GitHub MCP Server | Glama, accessed on April 15, 2025, https://glama.ai/mcp/servers/bnif4b1rh6/blob/master/docs/mcp-documentation.md
106. InitializeRequest in mcp_attr::schema - Rust - Docs.rs, accessed on April 15, 2025, https://docs.rs/mcp-attr/latest/mcp_attr/schema/struct.InitializeRequest.html
107. Transports - Model Context Protocol Specification, accessed on April 15, 2025, https://spec.modelcontextprotocol.io/specification/2025-03-26/basic/transports/
108. riza-io/mcp-go: Build Model Context Protocol (MCP) servers in Go - GitHub, accessed on April 15, 2025, https://github.com/riza-io/mcp-go
109. InitializeResult in mmcp_protocol::mcp - Rust - Docs.rs, accessed on April 15, 2025, https://docs.rs/mmcp-protocol/latest/mmcp_protocol/mcp/struct.InitializeResult.html
110. InitializeResult in mcp_attr::schema - Rust - Docs.rs, accessed on April 15, 2025, https://docs.rs/mcp-attr/latest/mcp_attr/schema/struct.InitializeResult.html
111. A type-safe implementation of the official Model Context Protocol (MCP) schema in Rust. - GitHub, accessed on April 15, 2025, https://github.com/rust-mcp-stack/rust-mcp-schema

112. Model Context Protocol (MCP): Lets Implement an MCP server in Go, accessed on April 15, 2025, https://prasanthmj.github.io/ai/mcp-go/
113. Gemini MCP Server | Glama, accessed on April 15, 2025, https://glama.ai/mcp/servers/bq7qhwa4y0/blob/main/src/protocol.ts
114. rust-mcp-sdk - Lib.rs, accessed on April 15, 2025, https://lib.rs/crates/rust-mcp-sdk
115. Lifecycle - Specification - Model Context Protocol, accessed on April 15, 2025, https://spec.modelcontextprotocol.io/specification/2024-11-05/basic/lifecycle/
116. InitializedNotification in mcp_attr::schema - Rust - Docs.rs, accessed on April 15, 2025, https://docs.rs/mcp-attr/latest/mcp_attr/schema/struct.InitializedNotification.html
117. MCP Web Tools Server - Glama, accessed on April 15, 2025, https://glama.ai/mcp/servers/cn5q63wd5r/blob/main/docs/05-communication-protocols.md
118. mcp package - github.com/loopwork-ai/emcee/mcp - Go Packages, accessed on April 15, 2025, https://pkg.go.dev/github.com/loopwork-ai/emcee/mcp
119. Error Handling — hermes_mcp v0.3.6 - HexDocs, accessed on April 15, 2025, https://hexdocs.pm/hermes_mcp/error_handling.html
120. JsonRpcError in async_mcp::transport - Rust - Docs.rs, accessed on April 15, 2025, https://docs.rs/async-mcp/latest/async_mcp/transport/struct.JsonRpcError.html
121. [BUG] MCP Server Integration: protocolVersion validation error with stdio servers · Issue #768 · anthropics/claude-code - GitHub, accessed on April 15, 2025, https://github.com/anthropics/claude-code/issues/768
122. ida-pro-mcp/mcp-plugin.py at main - GitHub, accessed on April 15, 2025, https://github.com/mrexodia/ida-pro-mcp/blob/main/mcp-plugin.py
123. A2A Client MCP Server | Glama, accessed on April 15, 2025, https://glama.ai/mcp/servers/@tesla0225/mcp-a2a/blob/main/a2a-client.ts
124. What is the Model Context Protocol (MCP)? A Complete Guide - Treblle Blog, accessed on April 15, 2025, https://blog.treblle.com/model-context-protocol-guide/
125. Messages - Specification - Model Context Protocol, accessed on April 15, 2025, https://spec.modelcontextprotocol.io/specification/2024-11-05/basic/messages/
126. mcp kotlin server does not process JSONRPCRequest with null param, but python sdk may send with null param · Issue #42 · modelcontextprotocol/kotlin-sdk - GitHub, accessed on April 15, 2025, https://github.com/modelcontextprotocol/kotlin-sdk/issues/42
127. mcp-framework/src/transports/http/server.ts at main - GitHub, accessed on April 15, 2025, https://github.com/QuantGeekDev/mcp-framework/blob/main/src/transports/http/server.ts
128. github.com/virgoC0der/go-mcp v0.1.3-0.20250409030620-629c17037678 on Go - Libraries.io - security & maintenance data for open source software, accessed on April 15, 2025,

https://libraries.io/go/github.com%2FvirgoC0der%2Fgo-mcp
129. A framework for writing MCP (Model Context Protocol) servers in Typescript - GitHub, accessed on April 15, 2025, https://github.com/QuantGeekDev/mcp-framework
130. Bug Report: MCP SSE Server Connects but Cursor Shows "No tools available" Despite Server Responding Correctly, accessed on April 15, 2025, https://forum.cursor.com/t/bug-report-mcp-sse-server-connects-but-cursor-shows-no-tools-available-despite-server-responding-correctly/77126
131. mssql-mcp-server/server.mjs at main - GitHub, accessed on April 15, 2025, https://github.com/dperussina/mssql-mcp-server/blob/main/server.mjs
132. ListPromptsRequestParams in mcp_attr::schema - Rust - Docs.rs, accessed on April 15, 2025, https://docs.rs/mcp-attr/latest/mcp_attr/schema/struct.ListPromptsRequestParams.html
133. langchaingo-mcp-adapter/adapter_test.go at main - GitHub, accessed on April 15, 2025, https://github.com/i2y/langchaingo-mcp-adapter/blob/main/adapter_test.go
134. straw_mcp - Dart API docs - Pub.dev, accessed on April 15, 2025, https://pub.dev/documentation/straw_mcp/latest/
135. mcp_dart library - Dart API - Pub.dev, accessed on April 15, 2025, https://pub.dev/documentation/mcp_dart/latest/mcp_dart
136. McpServer class - mcp_dart library - Dart API - pub.dev, accessed on April 15, 2025, https://pub.dev/documentation/mcp_dart/latest/mcp_dart/McpServer-class.html
137. ListPromptsResult in mcp_schema - Rust - Docs.rs, accessed on April 15, 2025, https://docs.rs/mcp-schema/latest/mcp_schema/struct.ListPromptsResult.html
138. ListPromptsResult in mcp_sdk_rs::types - Rust - Docs.rs, accessed on April 15, 2025, https://docs.rs/mcp-sdk-rs/latest/mcp_sdk_rs/types/struct.ListPromptsResult.html
139. No tools, no resources listed in MCP Servers in Cline · Issue #1272 - GitHub, accessed on April 15, 2025, https://github.com/cline/cline/issues/1272
140. ListResourcesRequest in mcp_core::resource - Rust - Docs.rs, accessed on April 15, 2025, https://docs.rs/mcp-core/latest/mcp_core/resource/struct.ListResourcesRequest.html
141. MCP connections disappear on their own : r/CLine - Reddit, accessed on April 15, 2025, https://www.reddit.com/r/CLine/comments/1j70s92/mcp_connections_disappear_on_their_own/
142. JSON Canvas MCP Server | Glama, accessed on April 15, 2025, https://glama.ai/mcp/servers/ajj64p9c3i/blob/main/simple_mcp_server.py
143. Resources - Model Context Protocol, accessed on April 15, 2025, https://modelcontextprotocol.io/docs/concepts/resources

144. Reference: MastraMCPClient | Tool Discovery | Mastra Docs, accessed on April 15, 2025, https://mastra.ai/docs/reference/tools/client
145. mcp_client_rs - crates.io: Rust Package Registry, accessed on April 15, 2025, https://crates.io/crates/mcp_client_rs
146. ListToolsResult in mcp_attr::schema - Rust - Docs.rs, accessed on April 15, 2025, https://docs.rs/mcp-attr/latest/mcp_attr/schema/struct.ListToolsResult.html
147. mcp-agent/src/mcp_agent/agents/agent.py at main - GitHub, accessed on April 15, 2025, https://github.com/lastmile-ai/mcp-agent/blob/main/src/mcp_agent/agents/agent.py
148. stephentoub/mcp: The official C# SDK for Model Context Protocol servers and clients, maintained by Microsoft - GitHub, accessed on April 15, 2025, https://github.com/stephentoub/mcp
149. camel.toolkits.mcp_toolkit — CAMEL 0.2.44 documentation, accessed on April 15, 2025, https://docs.camel-ai.org/_modules/camel/toolkits/mcp_toolkit.html
150. mcp-client/README.md at main - GitHub, accessed on April 15, 2025, https://github.com/punkpeye/mcp-client/blob/main/README.md
151. MCPD log level explanation - DevCentral - F5, accessed on April 15, 2025, https://community.f5.com/discussions/technicalforum/mcpd-log-level-explanation/84260
152. SetLevelRequest in mcp_attr::schema - Rust - Docs.rs, accessed on April 15, 2025, https://docs.rs/mcp-attr/latest/mcp_attr/schema/struct.SetLevelRequest.html
153. nutjs-windows-control - Glama, accessed on April 15, 2025, https://glama.ai/mcp/servers/@Cheffromspace/MCPControl/tree/master/docs
154. MCP Client Boot Starters | Spring AI1.0.0-SNAPSHOT中文文档, accessed on April 15, 2025, https://www.spring-doc.cn/spring-ai/1.0.0-SNAPSHOT/api_mcp_mcp-client-boot-starter-docs.html
155. Merge pull request #2 from ThinkInAIXYZ/feat/core feat(core): 简单的定义& gateway 实现· 8888/github.com/ThinkInAIXYZ/go-mcp - CNB, accessed on April 15, 2025, https://cnb.cool/8888/github.com/ThinkInAIXYZ/go-mcp/-/commit/81d1a9544237c1ba45b0cced074e84c01f15eb1f
156. mcp - Haystack Documentation - Deepset, accessed on April 15, 2025, https://docs.haystack.deepset.ai/v2.8/reference/integrations-mcp
157. McpError class - mcp_server library - Dart API - Pub.dev, accessed on April 15, 2025, https://pub.dev/documentation/mcp_server/latest/mcp_server/McpError-class.html
158. McpError: MCP error -32603: keyValidator._parse is not a function #291 - GitHub, accessed on April 15, 2025, https://github.com/modelcontextprotocol/typescript-sdk/issues/291
159. I'm getting MCP error with the latest update - #38 by deanrie - Bug Reports - Cursor Forum, accessed on April 15, 2025,

https://forum.cursor.com/t/im-getting-mcp-error-with-the-latest-update/72733/38
160. Expected manner to capture error details during startup · Issue #153 · modelcontextprotocol/typescript-sdk - GitHub, accessed on April 15, 2025, https://github.com/modelcontextprotocol/typescript-sdk/issues/153
161. MCP Error: "could not connect to MCP server" : r/ClaudeAI - Reddit, accessed on April 15, 2025, https://www.reddit.com/r/ClaudeAI/comments/1h18ep6/mcp_error_could_not_connect_to_mcp_server/
162. How to fix the error when decompiling MCP 9.18? - Stack Overflow, accessed on April 15, 2025, https://stackoverflow.com/questions/68255138/how-to-fix-the-error-when-decompiling-mcp-9-18
163. [Solved] MCP error | SpigotMC - High Performance Minecraft Community, accessed on April 15, 2025, https://www.spigotmc.org/threads/solved-mcp-error.402881/
164. MCP error. Trying to make a minecraft client - SpigotMC, accessed on April 15, 2025, https://www.spigotmc.org/threads/mcp-error-trying-to-make-a-minecraft-client.334057/
165. MCP Error Message : r/Morrowind - Reddit, accessed on April 15, 2025, https://www.reddit.com/r/Morrowind/comments/ofmzqh/mcp_error_message/
166. MCP Notify Server – A Model Context Protocol service that sends desktop notifications and alert sounds when AI agent tasks are completed, integrating with various LLM clients like Claude Desktop and Cursor. - Reddit, accessed on April 15, 2025, https://www.reddit.com/r/mcp/comments/1jd6j7x/mcp_notify_server_a_model_context_protocol/
167. Motor Carrier Permits - California DMV, accessed on April 15, 2025, https://www.dmv.ca.gov/portal/vehicle-industry-services/motor-carrier-services-mcs/motor-carrier-permits/
168. MCP Security Notification: Tool Poisoning Attacks - Invariant Labs, accessed on April 15, 2025, https://invariantlabs.ai/blog/mcp-security-notification-tool-poisoning-attacks
169. 5n7/mcp-notify: A simple MCP server for sending desktop notifications. - GitHub, accessed on April 15, 2025, https://github.com/5n7/mcp-notify
170. 02 - MCP Notifcation Overview - YouTube, accessed on April 15, 2025, https://m.youtube.com/watch?v=I6KmUtaGrh0
171. MCP 5 Minute Moments - Mass.gov, accessed on April 15, 2025, https://www.mass.gov/lists/mcp-5-minute-moments
172. MCP Amendments | Mass.gov, accessed on April 15, 2025, https://www.mass.gov/lists/mcp-amendments
173. 2024 Amendments to the Massachusetts Contingency Plan, effective March 1 2024, accessed on April 15, 2025, https://www.lspa.org/index.php?option=com_dailyplanetblog&view=entry&year=2

024&month=02&day=29&id=442:2024-amendments-to-the-massachusetts-contingency-plan-effective-march-1-2024
174. Reportable Conditions under the Massachusetts Contingency Plan, accessed on April 15, 2025, https://www.omnieg.com/2017/06/28/reportable-conditions-under-the-massachusetts-contingency-plan/
175. MQTT 5.0 Packet Explained 04: PINGREQ & PINGRESP | EMQ - EMQX, accessed on April 15, 2025, https://www.emqx.com/en/blog/mqtt-5-0-control-packets-04-pingreq-pingresp
176. Enabling MCP PING Commands through the EVLAN Connection - Libra 6400/8400 System Implementation Guide - Product Support, accessed on April 15, 2025, https://public.support.unisys.com/c71/docs/Libra6400-8400-1.0/82057456-000/section-000001662.html
177. Enabling MCP PING Commands through the EVLAN Connection, accessed on April 15, 2025, https://public.support.unisys.com/mds/docs/Release5.0/82050246-009/WebHelp/82050246-009/section-000014273.html
178. WL1801: RX packets delay and loss trouble - Wi-Fi forum - TI E2E, accessed on April 15, 2025, https://e2e.ti.com/support/wireless-connectivity/wi-fi-group/wifi/f/wi-fi-forum/953598/wl1801-rx-packets-delay-and-loss-trouble
179. Progress - Specification - Model Context Protocol, accessed on April 15, 2025, https://spec.modelcontextprotocol.io/specification/2025-03-26/basic/utilities/progress/
180. [Protocol] Implement MCP progress notification support · Issue #12 · cloudwalk/hermes-mcp, accessed on April 15, 2025, https://github.com/cloudwalk/hermes-mcp/issues/12
181. Key Changes - Model Context Protocol, accessed on April 15, 2025, https://modelcontextprotocol.io/specification/2025-03-26/changelog
182. Quarkus MCP Server - Quarkiverse Documentation, accessed on April 15, 2025, https://docs.quarkiverse.io/quarkus-mcp-server/dev/index.html
183. Whisper with Claude through MCP : r/modelcontextprotocol - Reddit, accessed on April 15, 2025, https://www.reddit.com/r/modelcontextprotocol/comments/1h88pmj/whisper_with_claude_through_mcp/
184. ProgressNotification in rust_mcp_schema - Rust - Docs.rs, accessed on April 15, 2025, https://docs.rs/rust-mcp-schema/latest/rust_mcp_schema/struct.ProgressNotification.html
185. Git Spice Help MCP Server | Glama, accessed on April 15, 2025, https://glama.ai/mcp/servers/@sach999/git-spice-help-mcp?
186. Prompts - Model Context Protocol Specification, accessed on April 15, 2025, https://spec.modelcontextprotocol.io/specification/2024-11-05/server/prompts/
187. buecking/hs-mcp: Haskell Client/Server implantation of MCP (Model Context

Protocol), accessed on April 15, 2025, https://github.com/buecking/hs-mcp
188. Google Maps MCP Server | Glama, accessed on April 15, 2025, https://glama.ai/mcp/servers/@Moe03/mcp-hello-world/tree/main/src
189. Model Context Protocol (MCP) Explained: The New Standard for AI Tools? - YouTube, accessed on April 15, 2025, https://www.youtube.com/watch?v=rmphqjsc4Po
190. cremich/promptz-mcp: MCP Server implementation for ... - GitHub, accessed on April 15, 2025, https://github.com/cremich/promptz-mcp
191. protocol package - github.com/go-go-golems/go-go-mcp/pkg, accessed on April 15, 2025, https://pkg.go.dev/github.com/go-go-golems/go-go-mcp/pkg/protocol
192. ReadResourceRequest in mcp_attr::schema - Rust - Docs.rs, accessed on April 15, 2025, https://docs.rs/mcp-attr/latest/mcp_attr/schema/struct.ReadResourceRequest.html
193. The official Kotlin SDK for Model Context Protocol servers and clients. Maintained in collaboration with JetBrains - GitHub, accessed on April 15, 2025, https://github.com/modelcontextprotocol/kotlin-sdk
194. JetBrains/mcp-kotlin-sdk: Kotlin implementation of the Model Context Protocol (MCP), accessed on April 15, 2025, https://github.com/JetBrains/mcp-kotlin-sdk
195. MCP Resources | GenAIScript - Microsoft Open Source, accessed on April 15, 2025, https://microsoft.github.io/genaiscript/blog/mcp-resources/
196. Resources - Model Context Protocol Specification, accessed on April 15, 2025, https://spec.modelcontextprotocol.io/specification/2024-11-05/server/resources/
197. Model Context Protocol - Cursor, accessed on April 15, 2025, https://docs.cursor.com/context/model-context-protocol
198. MCP Resources vs Tools : r/ClaudeAI - Reddit, accessed on April 15, 2025, https://www.reddit.com/r/ClaudeAI/comments/1jso42a/mcp_resources_vs_tools/
199. MCP Resources | Model Context Protocol | Python - YouTube, accessed on April 15, 2025, https://www.youtube.com/watch?v=M3MnRc77yO8
200. ResourceUpdatedNotification in mcp_attr::schema - Rust - Docs.rs, accessed on April 15, 2025, https://docs.rs/mcp-attr/latest/mcp_attr/schema/struct.ResourceUpdatedNotification.html
201. mcp-swift-sdk - Swift Package Index, accessed on April 15, 2025, https://swiftpackageindex.com/loopwork-ai/mcp-swift-sdk
202. specification/schema/2025-03-26/schema.json at main - GitHub, accessed on April 15, 2025, https://github.com/modelcontextprotocol/specification/blob/main/schema/2025-03-26/schema.json
203. mcp-swift-sdk – Swift Package Index, accessed on April 15, 2025, https://swiftpackageindex.com/modelcontextprotocol/swift-sdk
204. OpenAI just announced 'Tasks' which is basically a prompt ... - Reddit, accessed on April 15, 2025,

https://www.reddit.com/r/mcp/comments/1i1yyx7/openai_just_announced_tasks_which_is_basically_a/
205. A Journey from AI to LLMs and MCP - 7 - Under the Hood — The Architecture of MCP and Its Core Components - DEV Community, accessed on April 15, 2025, https://dev.to/alexmercedcoder/a-journey-from-ai-to-llms-and-mcp-7-under-the-hood-the-architecture-of-mcp-and-its-core-4jme
206. mohamedahmed01/laravel-mcp-sdk: Laravel Based Implementation for Model Context Protocol - GitHub, accessed on April 15, 2025, https://github.com/mohamedahmed01/laravel-mcp-sdk
207. MCP + image support - Bug Reports - Cursor - Community Forum, accessed on April 15, 2025, https://forum.cursor.com/t/mcp-image-support/48801
208. Examining the role of within functional area resource–capability complementarity in achieving customer and product-based performance outcomes | Request PDF - ResearchGate, accessed on April 15, 2025, https://www.researchgate.net/publication/241733137_Examining_the_role_of_within_functional_area_resource-capability_complementarity_in_achieving_customer_and_product-based_performance_outcomes
209. ProgressNotificationParams in mcp_schema - Rust - Docs.rs, accessed on April 15, 2025, https://docs.rs/mcp-schema/latest/mcp_schema/struct.ProgressNotificationParams.html
210. TimHolden_figma-mcp-server/README.md at main · MCP-Mirror, accessed on April 15, 2025, https://github.com/MCP-Mirror/TimHolden_figma-mcp-server/blob/main/README.md
211. Roots - Model Context Protocol, accessed on April 15, 2025, https://modelcontextprotocol.io/specification/2025-03-26/client/roots
212. Roots - Model Context Protocol, accessed on April 15, 2025, https://modelcontextprotocol.io/docs/concepts/roots
213. solana-mcp-server/protocol.txt at main - GitHub, accessed on April 15, 2025, https://github.com/openSVM/solana-mcp-server/blob/main/protocol.txt
214. Sampling - Model Context Protocol, accessed on April 15, 2025, https://modelcontextprotocol.io/specification/2025-03-26/client/sampling
215. MCP-Mod (Multiple Comparisons Procedure - Modelling) Explained - Statsols, accessed on April 15, 2025, https://www.statsols.com/articles/mcp-mod
216. What is sampling for? : r/mcp - Reddit, accessed on April 15, 2025, https://www.reddit.com/r/mcp/comments/1hhllm3/what_is_sampling_for/
217. The New MCP on Sampling and Exposure Points - LSP Association, accessed on April 15, 2025, https://www.lspa.org/index.php?option=com_dailyplanetblog&view=entry&year=2024&month=01&day=07&id=434:the-new-mcp-on-sampling-and-exposure-points
218. MVP/GMVP Version 3 : General Purpose Monte Carlo Codes for Neutron and Photon Transport Calculations Based on Continuous Energy - JOPSS, accessed on April 15, 2025, https://jopss.jaea.go.jp/pdfdata/JAEA-Data-Code-2016-018.pdf

219. Fax: +49(0)4154 - 740 16 • E-mail: information@PKI-electronic.com Copyright even of extracts b, accessed on April 15, 2025, https://pki-electronic.com/wp-content/uploads/2022/04/New-Catalogue-2022.pdf
220. Aerosol, Cloud, Ecosystems (ACE) Final Study Report - NASA GMAO, accessed on April 15, 2025, https://gmao.gsfc.nasa.gov/pubs/docs/da%20Silva1247.pdf
221. Applied Geophysics, accessed on April 15, 2025, https://ia601503.us.archive.org/34/items/in.ernet.dli.2015.118629/2015.118629.Applied-Geophysics_text.pdf
222. Chronic tonsillitis is not associated with beta defensin 1 gene polymorphisms in Turkish population | Request PDF - ResearchGate, accessed on April 15, 2025, https://www.researchgate.net/publication/272422297_Chronic_tonsillitis_is_not_associated_with_beta_defensin_1_gene_polymorphisms_in_Turkish_population
223. For Server Developers - Model Context Protocol, accessed on April 15, 2025, https://modelcontextprotocol.io/quickstart/server
224. MCP Overview - Cline Documentation, accessed on April 15, 2025, https://docs.cline.bot/mcp-servers/mcp
225. MCP Marketplace - Cline Documentation, accessed on April 15, 2025, https://docs.cline.bot/mcp-servers/mcp-quickstart
226. MCP tools - Agent Development Kit - Google, accessed on April 15, 2025, https://google.github.io/adk-docs/tools/mcp-tools/
227. Model Context Protocol (MCP): 8 MCP Servers Every Developer Should Try!, accessed on April 15, 2025, https://dev.to/pavanbelagatti/model-context-protocol-mcp-8-mcp-servers-every-developer-should-try-5hm2
228. ServerCapabilities in mcp_attr::schema - Rust - Docs.rs, accessed on April 15, 2025, https://docs.rs/mcp-attr/latest/mcp_attr/schema/struct.ServerCapabilities.html
229. MCP Notify Server | Glama, accessed on April 15, 2025, https://glama.ai/mcp/servers/@Cactusinhand/mcp_server_notify
230. Cloudinary MCP Server | Glama, accessed on April 15, 2025, https://glama.ai/mcp/servers/zjiw1ry8ly/blob/master/documentation/mcp-llm-guide.md
231. Exploring the Model Context Protocol with Deno 2 and Playwright - Shruggingface, accessed on April 15, 2025, https://www.shruggingface.com/microblog/2024/12/02/exploring-the-model-context-protocol-with-deno-2-and-playwright
232. ServerNotification in mcp_schema - Rust - Docs.rs, accessed on April 15, 2025, https://docs.rs/mcp-schema/latest/mcp_schema/enum.ServerNotification.html
233. server package - github.com/dceluis/mcp-go/server - Go Packages, accessed on April 15, 2025, https://pkg.go.dev/github.com/dceluis/mcp-go/server
234. keystone/server/notification - salt-models/mcp-virtual-aio - Gitiles, accessed on April 15, 2025,

https://gerrit.mcp.mirantis.com/plugins/gitiles/salt-models/mcp-virtual-aio/+/716cc6744ccca8a90393b4c1ccab8d74a46caca9/keystone/server/notification?autodive=0%2F%2F%2F/
235. heat/server/notification - salt-models/mcp-virtual-aio - Gitiles, accessed on April 15, 2025, https://gerrit.mcp.mirantis.com/plugins/gitiles/salt-models/mcp-virtual-aio/+/3ab784b7fda41a5d38b5c0d7046aa889bd921a6b/heat/server/notification?autodive=0%2F%2F%2F%2F%2F%2F%2F/
236. What's MCP all about? Comparing MCP with LLM function calling - Neon, accessed on April 15, 2025, https://neon.tech/blog/mcp-vs-llm-function-calling
237. Connecting to a Model Context Protocol (MCP) Server from Java using LangChain4j, accessed on April 15, 2025, https://blog.marcnuri.com/connecting-to-mcp-server-with-langchain4j
238. yaxingson/mcp-server-request - GitHub, accessed on April 15, 2025, https://github.com/yaxingson/mcp-server-request
239. A Complete Guide to the Model Context Protocol (MCP) in 2025 - Keywords AI, accessed on April 15, 2025, https://www.keywordsai.co/blog/introduction-to-mcp
240. Hanzo MCP | Glama, accessed on April 15, 2025, https://glama.ai/mcp/servers/@hanzoai/mcp
241. Using an MCP SSE Server with Parakeet - Philippe Charrière's Blog, accessed on April 15, 2025, https://k33g.hashnode.dev/using-an-mcp-sse-server-with-parakeet
242. mcp-agent - PyPI, accessed on April 15, 2025, https://pypi.org/project/mcp-agent/
243. How to Build and Publish an MCP Server - A Detailed Guide - YouTube, accessed on April 15, 2025, https://www.youtube.com/watch?v=CiArUs_2jm4
244. dend/entra-id-mcp-server-session-demo - GitHub, accessed on April 15, 2025, https://github.com/dend/entra-id-mcp-server-session-demo
245. Model Context Protocol (MCP) and Amazon Bedrock - Community.aws, accessed on April 15, 2025, https://community.aws/content/2uFvyCPQt7KcMxD9ldsJyjZM1Wp/model-context-protocol-mcp-and-amazon-bedrock
246. gala-soso/mcp-rules - Explore custom AI code assistants, accessed on April 15, 2025, https://hub.continue.dev/gala-soso/mcp-rules
247. session package - github.com/Hirocloud/mcp-go/server/session - Go Packages, accessed on April 15, 2025, https://pkg.go.dev/github.com/Hirocloud/mcp-go/server/session
248. FastMCP Multi-Tenancy | Glama, accessed on April 15, 2025, https://glama.ai/mcp/servers/@timothywangdev/McpToolKit/tree/master/src
249. Model Context Protocol (MCP) Clearly Explained : r/LLMDevs - Reddit, accessed on April 15, 2025, https://www.reddit.com/r/LLMDevs/comments/1jbqegg/model_context_protocol_mcp_clearly_explained/
250. Docker · Issue #879 · modelcontextprotocol/servers - GitHub, accessed on

April 15, 2025, https://github.com/modelcontextprotocol/servers/issues/879
251. The ULTIMATE Guide to Building Your Own MCP Servers (Free Template) - YouTube, accessed on April 15, 2025, https://www.youtube.com/watch?v=lbyPJqCI-tw
252. org.http4k.mcp.server.session, accessed on April 15, 2025, https://www.http4k.org/api/http4k-mcp-sdk/org.http4k.mcp.server.session/index.html
253. Logging - Model Context Protocol, accessed on April 15, 2025, https://modelcontextprotocol.io/specification/2025-03-26/server/utilities/logging
254. Microsoft for Developers, accessed on April 15, 2025, https://devblogs.microsoft.com/feed
255. 从零开始学MCP | 什么是MCP？MCP 核心概念详细解析！ - CSDN博客, accessed on April 15, 2025, https://blog.csdn.net/m0_56255097/article/details/147149818
256. LangChain MCP Adapters - GitHub, accessed on April 15, 2025, https://github.com/langchain-ai/langchain-mcp-adapters
257. Tools - Hugging Face, accessed on April 15, 2025, https://huggingface.co/docs/smolagents/v1.4.1/reference/tools
258. Server - PydanticAI, accessed on April 15, 2025, https://ai.pydantic.dev/mcp/server/
259. Unleashing the Power of Model Context Protocol (MCP): A Game-Changer in AI Integration, accessed on April 15, 2025, https://techcommunity.microsoft.com/blog/educatordeveloperblog/unleashing-the-power-of-model-context-protocol-mcp-a-game-changer-in-ai-integrat/4397564
260. Using MCP Servers with OpenRouter, accessed on April 15, 2025, https://openrouter.ai/docs/use-cases/mcp-servers
261. MCP Servers - OpenAI Agents SDK, accessed on April 15, 2025, https://openai.github.io/openai-agents-python/ref/mcp/server/
262. Create a Custom MCP Client To Learn and Manage Your AWS Services and Resources, accessed on April 15, 2025, https://dev.to/aws/create-a-custom-mcp-client-to-learn-and-manage-your-aws-services-and-resources-2omc
263. How to use Anthropic MCP Server with open LLMs, OpenAI or Google Gemini - Philschmid, accessed on April 15, 2025, https://www.philschmid.de/mcp-example-llama
264. Sampling - Model Context Protocol, accessed on April 15, 2025, https://modelcontextprotocol.io/specification/2024-11-05/client/sampling
265. Yet another MCP client for Bedrock - Community.aws, accessed on April 15, 2025, https://community.aws/content/2v8m09JtrzZ4VVsOD4mBflVZqq3/yet-another-mcp-client-for-bedrock
266. MCP Electronic Module | Pull Stations | Manual Initiating Devices | Voice & Notification, accessed on April 15, 2025, https://buildings.honeywell.com/us/en/products/by-category/fire-life-safety/manu

al-call-points-pull-stations-and-panic-buttons/manual-call-points-pull-stations/mcp-compact-small-addressable-with-resettable-element
267. Model Context Protocol (MCP): A Guide With Demo Project - DataCamp, accessed on April 15, 2025, https://www.datacamp.com/tutorial/mcp-model-context-protocol
268. Handle remote tool calling with Model Context Protocol - IBM Developer, accessed on April 15, 2025, https://developer.ibm.com/tutorials/awb-handle-remote-tool-calling-model-context-protocol/
269. Resources - Model Context Protocol, accessed on April 15, 2025, https://modelcontextprotocol.io/specification/2025-03-26/server/resources
270. UnsubscribeRequest in mcp_attr::schema - Rust - Docs.rs, accessed on April 15, 2025, https://docs.rs/mcp-attr/latest/mcp_attr/schema/struct.UnsubscribeRequest.html
271. SubscribeRequest (AWS SDK for Java - 1.12.782), accessed on April 15, 2025, https://docs.aws.amazon.com/AWSJavaSDK/latest/javadoc/com/amazonaws/services/codestarnotifications/model/SubscribeRequest.html
272. Documentation for MCP Drupal Module, accessed on April 15, 2025, https://mcp-77a54f.pages.drupalcode.org/
273. f/mcptools: A command-line interface for interacting with MCP (Model Context Protocol) servers using both stdio and HTTP transport. - GitHub, accessed on April 15, 2025, https://github.com/f/mcptools
274. Use MCP servers in VS Code (Preview), accessed on April 15, 2025, https://code.visualstudio.com/docs/copilot/chat/mcp-servers
275. Open-Source MCP tools - Glama, accessed on April 15, 2025, https://glama.ai/mcp/tools
276. Tools - Model Context Protocol, accessed on April 15, 2025, https://modelcontextprotocol.io/docs/concepts/tools
277. Specification - Model Context Protocol, accessed on April 15, 2025, https://modelcontextprotocol.io/specification/2025-03-26
278. Documentation Crawler & MCP Server - Glama, accessed on April 15, 2025, https://glama.ai/mcp/servers/@alizdavoodi/MCPDocSearch/tree/main/mcp_server
279. modelcontextprotocol/csharp-sdk: The official C# SDK for Model Context Protocol servers and clients. Maintained in collaboration with Microsoft. - GitHub, accessed on April 15, 2025, https://github.com/modelcontextprotocol/csharp-sdk
280. Tools - Model Context Protocol Specification, accessed on April 15, 2025, https://spec.modelcontextprotocol.io/specification/2024-11-05/server/tools/
281. Decision Support Tool for Risk Assessment & Maneuver Planning in Collision Avoidance Alexander Ryan Lead Aerospace Engineer, accessed on April 15, 2025, https://ntrs.nasa.gov/api/citations/20220012070/downloads/FINAL%20ISG%20AMOS%20Paper-Decision%20Support%20Tool%20for%20Risk%20Assessment%20%26%20Maneuver%20Planning%20for%20CA.pdf
282. DynamoDB MCP Server | Glama, accessed on April 15, 2025, https://glama.ai/mcp/servers/3voqtftc3c/blob/main/tests/test_server.py
283. [Server] Add resources subscription and unsubscription support · Issue #14 ·

cloudwalk/hermes-mcp - GitHub, accessed on April 15, 2025, https://github.com/cloudwalk/hermes-mcp/issues/14
284. GOES-18 SEISS MPS-LO Electron and Ion Microchannel Plate (MCP) HV Bias Calibration (impacts SEISS) - NOAA OSPO, accessed on April 15, 2025, https://www.ospo.noaa.gov/data/messages/2025/02/MSG_20250213_1932.html
285. Understanding the Model Context Protocol (MCP) - Philippe Charrière's Blog, accessed on April 15, 2025, https://k33g.hashnode.dev/understanding-the-model-context-protocol-mcp
286. What is Model Context Protocol? - Portkey, accessed on April 15, 2025, https://portkey.ai/blog/model-context-protocol-for-llm-appls