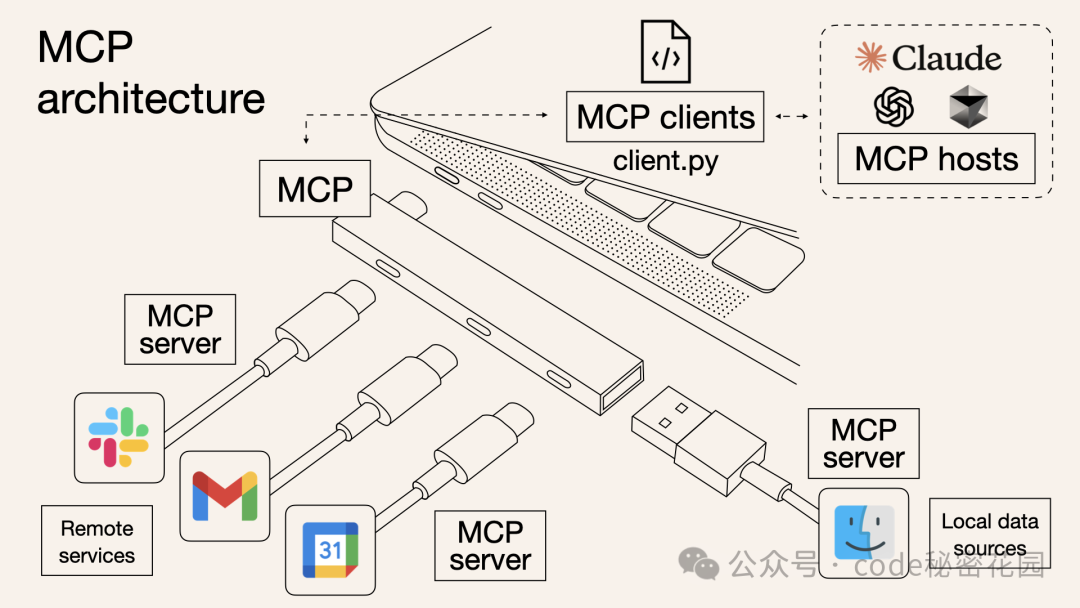
**MCP调研报告**

1. **什么是MCP（Model Context Protocol）**

MCP（Model Context Protocol，模型上下文协议） ，2024年11月底，由 Anthropic 推出的一种开放标准，旨在统一大型语言模型（LLM）与外部数据源和工具之间的通信协议。MCP 的主要目的在于解决当前 AI 模型因数据孤岛限制而无法充分发挥潜力的难题，MCP 使得 AI 应用能够安全地访问和操作本地及远程数据，为 AI 应用提供了连接万物的接口。



2. **发展历程**

**阶段一、传统大模型：**

为了让大模型等 AI 应用使用我们的数据，要么复制粘贴，要么上传下载,即使是最强大模型也会受到数据隔离的限制，形成信息孤岛，要做出更强大的模型，每个新数据源都需要自己重新定制实现，使真正互联的系统难以扩展，存在很多的局限性。



|  |
| --- |
| **传统大模型的局限性：**   1. 传统大模型受到数据隔离的限制 2. 大模型的核心功能是生成 token 和提供答案，因此它无法直接执行一些精细且需操作的具体任务，这也不是当前大模型的强项  * 比如调用其他服务的 API 帮你完成订餐和购票。 * 又比如比较 9.8 和 9.11 哪个大，或者计算 Strawberry 中到底有几个 r |

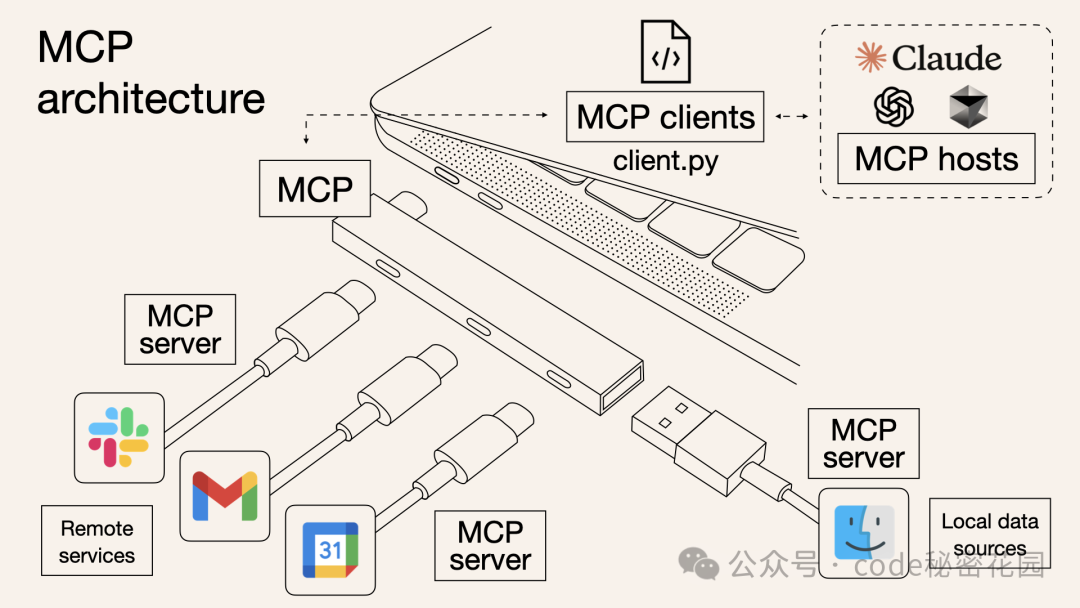
**阶段二、FunctionCall（OPEN AI 在 2023 提出）**

|  |  |
| --- | --- |
|  |  |

|  |
| --- |
| **智能体-工具**   1. 优势：解决了传统大模型的局限性（左图） 2. 劣势： 3. 依赖模型能力：模型本身需要进行过专门的 Function Call 调用微调 4. 各家模型不通用（右图）：   OPEN AI 最开始提出这项技术的时候，并没有想让它成为一项标准，所以虽然后续很多模型也支持了 Function Call 的调用，但是各自实现的方式都不太一样。  这也就意味着，如果我们要发开一个 Function Call 工具，需要对不同的模型进行适配，比如参数格式、触发逻辑、返回结构等等，这个成本是非常高的。 |

**阶段三、MCP（ Anthropic 在 2024 .11 提出）**

|  |  |
| --- | --- |
|  |  |



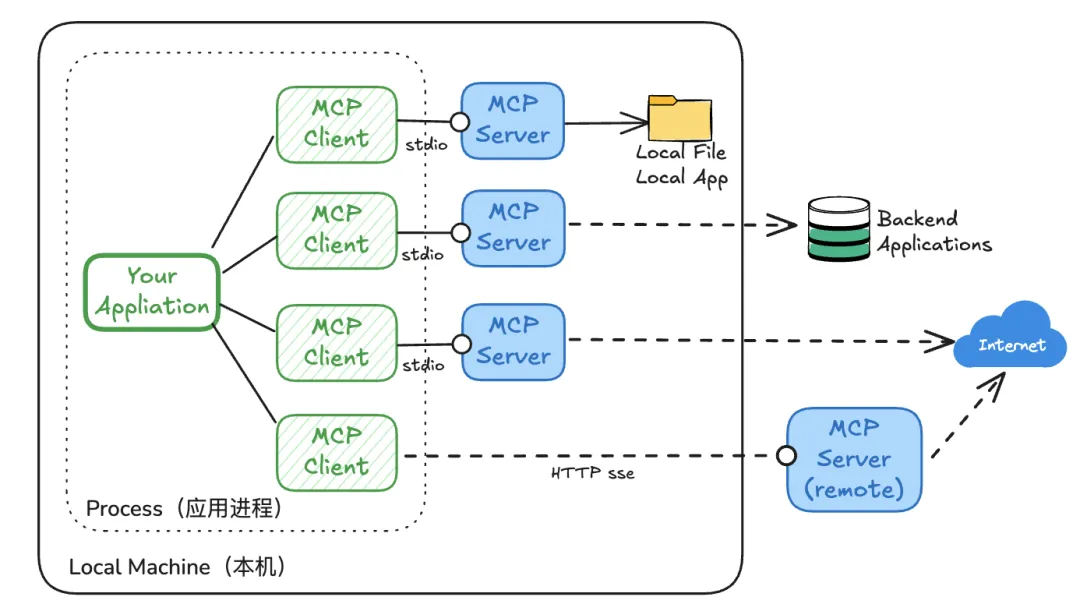
|  |
| --- |
| **MCP**   1. **核心贡献：统一规范**   就像是一个 “通用插头” 或者 “USB 接口”，制定了统一的规范，不管是连接数据库、第三方 API，还是本地文件等各种外部资源，都可以通过这个 “通用接口” 来完成，让 AI 模型与外部工具或数据源之间的交互更加标准化、可复用。   1. **带来的优势：**  * **LLM应用的简化：**不用适配各种私有协议，你只需要知道怎么连接MCP server * **LLM应用的快速扩展：**随时“插拔”新的MCP Server即可，一个不够就再来一个   MCP的提出者Anthropic旗下的Claude Desktop就可以通过这种方式扩充能力。   * **快速的适应变化：**想象下，如果一个外部资源的接口发生变化，只需要访问它的MCP Server做修改，所有的LLM应用就可无缝适应。 * **新的AI能力共享生态：**通过MCP Server的共享，新的LLM应用可以快速获得各种工具，形成了一种新的合作体系，提高整体效用。 |

**总结：FunctionCall和MCP的区别**



3. **核心架构**

**3.1 概念解析：**



* MCP 主机（MCP Hosts）：发起请求的 LLM 应用程序（例如 [Claude Desktop](https://zhida.zhihu.com/search?content_id=254488153&content_type=Article&match_order=1&q=Claude+Desktop&zhida_source=entity)、IDE 或 AI 工具）。
* MCP 客户端（MCP Clients）：由客户端LLM应用使用Client SDK创建并维护的一个**Server会话**。
* MCP 服务器（MCP Servers）：为Client 提供上下文、工具和 prompt 等信息的服务。可以部署在LLM应用本机，也可以远程部署（Remote）。

**3.2 工作流程：**

**[该类型的内容暂不支持下载]**

|  |
| --- |
| MCP 服务器（MCP Servers）   1. **提供哪些服务：**  * Tools：提供给LLM应用特别是Agent使用的工具。 * Resoures：提供给LLM应用一些额外的结构化数据。 * Prompts：提供给LLM应用的一些Prompt模板。比如你的应用是一个Chatbot，可以从MCP Server中取出这些模板，让使用者选择使用。  1. **使用什么语言创建？**   目前支持TS/Python/Java SDK来编写MCP Server。   1. **有哪些获取途径？**  * Github：[MCP Servers Repository](https://github.com/modelcontextprotocol/servers) 、[Awesome MCP Servers](https://github.com/punkpeye/awesome-mcp-servers) * 网站：   Sminthery : https://sminthery.com/  Pulse mcp : https://www.pulsemcp.com/  Awesome mcp servers : https://mcpservers.org/  mcp.so : https://mcp.so/  Galma.ai : https://glama.ai/mcp/servers  Cursor.directory : https://cursor.directory/ |

4. **使用方式**

**4.1 使用渠道**

* Cursor
* Claude Desktop
* Cline
* CherryStudio
* 代码调用
* 智能体平台（如dify、阿里百炼）

**4.2 使用示例**

以CherryStudio为例子

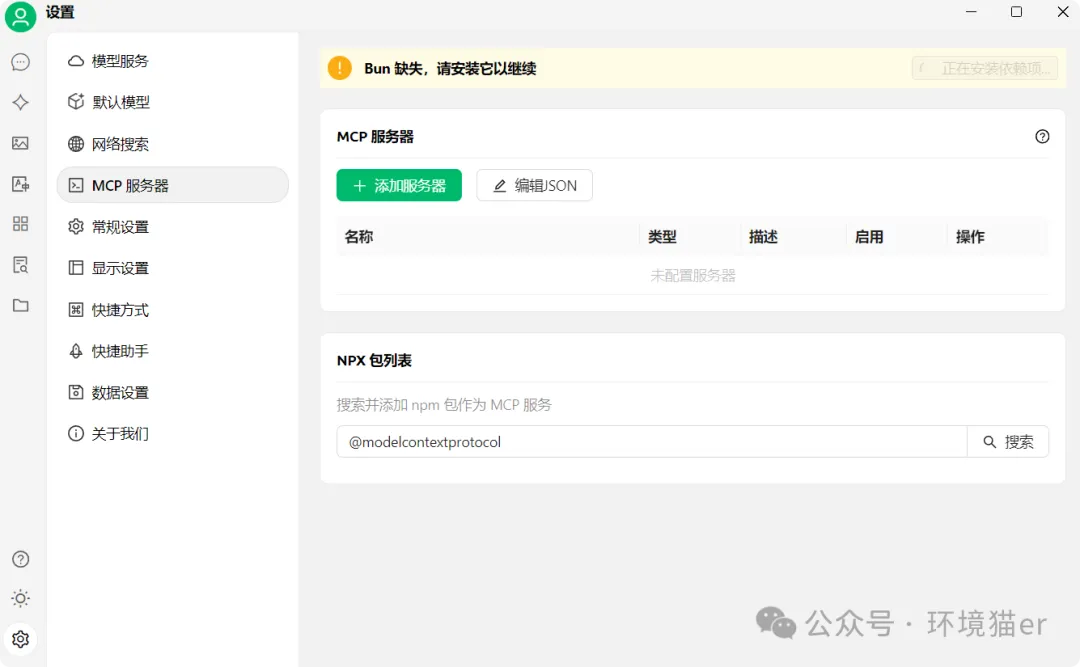
**4.2.1 为什么选择 Cherry Studio？**

Cherry Studio 是一款功能全面的 AI 桌面客户端，凭借其友好的用户界面和易用性，深受用户喜爱。近期，Cherry Studio 已开始支持 MCP 服务。本文将以 Cherry Studio 官方文档中提供的 fetch 工具为例，帮助用户快速理解并上手 MCP。同时，我们将通过举一反三的方式，演示如何使用 filesystem 工具操作文件夹，实现通过对话控制桌面文件。通过这些示例，旨在帮助用户了解 MCP 强大的扩展性。

**4.2.2 准备工作**

在开始之前，我们需要先安装以下工具：

1. **uv**
2. **bun**

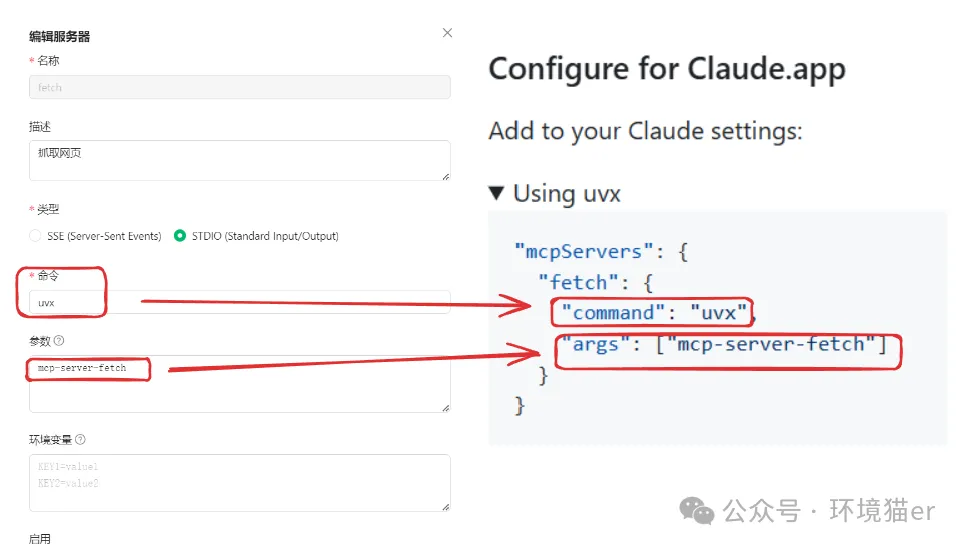


mcp工具添加

* **uv 是什么？**uv 是一个快速的 Python 包安装和依赖解析器。它可以显著提升 Python 项目的依赖管理速度。
* **Bun 是什么？**Bun 是一个快速的全面的 JavaScript 工具包，可以用来进行 JavaScript 和 TypeScript 项目的打包、测试和运行。

**4.2.3 配置 fetch Server**

为什么要选择 fetch 工具？因为它能够抓取网页信息，配置简单，方便我们理解 MCP 的配置和使用方法。



配置fetch

配置方法如下：

1. 打开 Cherry Studio，进入设置界面。
2. 找到 "MCP 服务器" 选项，点击 "添加服务器"。
3. 填写 MCP Server 的相关参数（参考链接）。

* **名称**：自定义一个名称，例如 fetch。
* **类型**：选择 STDIO。
* **命令**：填写 uvx。
* **参数**：填写 mcp-server-fetch。

**这些参数代表什么？为什么要这样填写？**

* **STDIO（标准输入/输出）**：在本地运行，可访问本机文件和应用程序，但需要配置 Python 和 NodeJS 环境。
* **SSE（服务器发送事件）**：在远程服务器运行，配置简单，但无法访问本地资源。

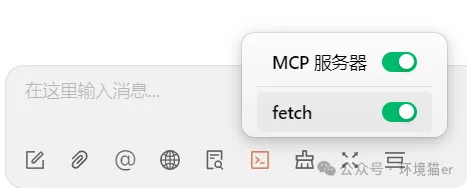
可以参考 fetch 工具在 GitHub 上的官方介绍。通过 uvx 安装的代码可以看到：

|  |
| --- |
| Plain Text "mcpServers": {  "fetch": {  "command": "uvx",  "args": ["mcp-server-fetch"]  } } |

Command 对应命令，args 对应参数。对于公开的 MCP 工具，我们通常可以这样找到相应的设置方法，后面我还会介绍如何查找更多公开的 MCP 服务器。

**4.2.4 使用方法**

1. 安装完服务器后，在对话界面点击启用 MCP 服务器。
2. 选择支持 MCP 的模型（带有扳手 🔧 标志）。
3. 就可以开始对话了！

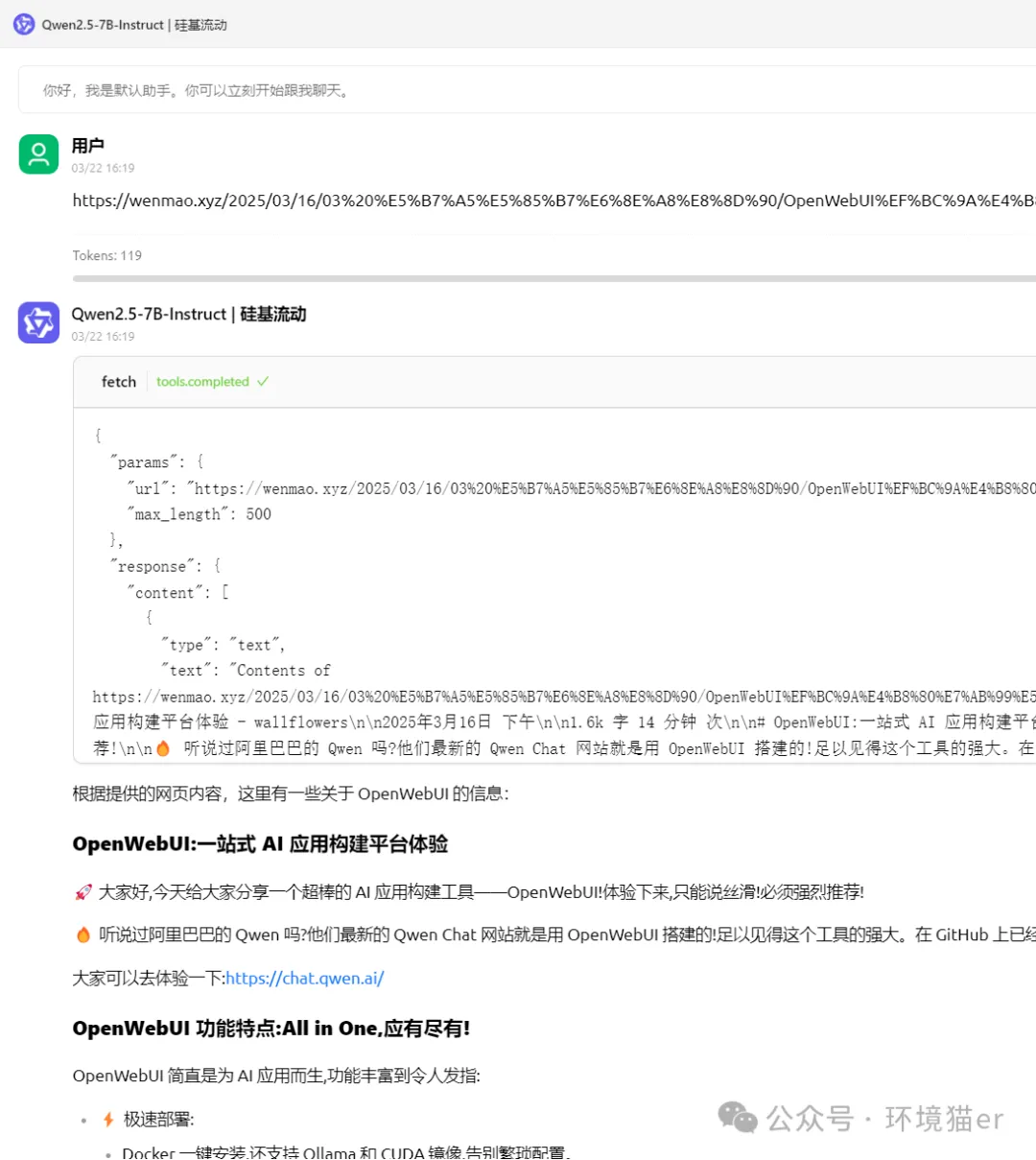


使用mcp功能



选择支持mcp的模型

**对话效果：**



对话效果

经过测试，即使是硅基流动免费的 Qwen 2.5-7 B-Instruct 模型，也可以成功抓取对应的网页内容。

**4.2.5 filesystem 服务**

学会了 fetch 工具的添加方法，我们就可以举一反三，添加更多的 MCP 工具。例如，filesystem 工具可以帮助我们操作本地文件。

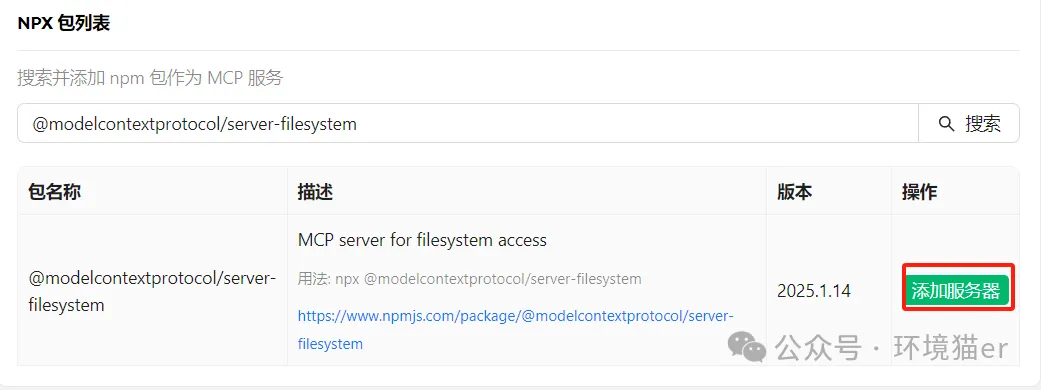
**filesystem 的主要功能**

**Filesystem 的关键功能包括：**

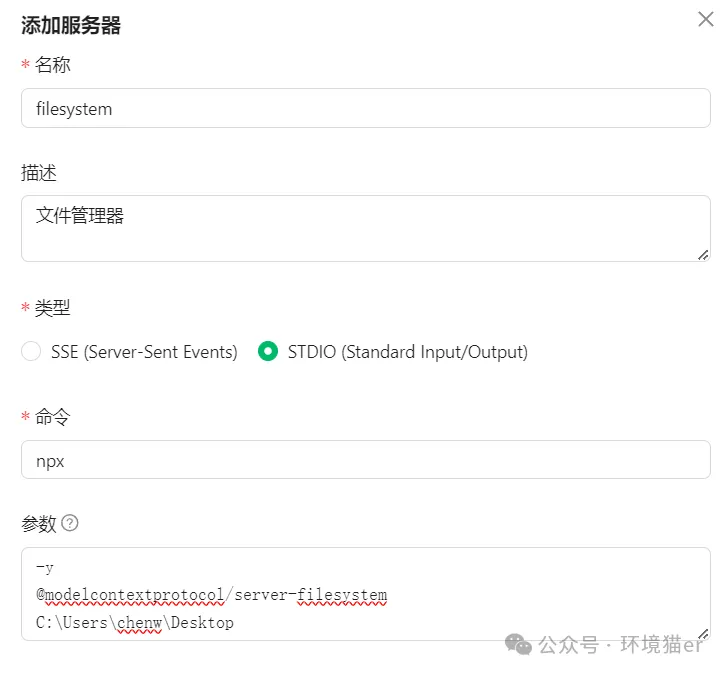
* **读/写文件和目录：**
* 这意味着 Filesystem 允许用户读取文件的内容，以及将新的内容写入文件。
* **创建、列出、删除或移动目录：**
* 用户可以使用 Filesystem 创建新的目录，查看现有目录的内容，删除不再需要的目录，以及将目录移动到新的位置。
* **使用模式匹配搜索文件：**
* Filesystem 允许用户使用通配符（例如，\*.txt）来搜索符合特定模式的文件。这使得查找文件变得更加方便。
* **检索详细的文件元数据：**
* 元数据指的是文件的属性，例如文件的大小、创建时间、修改时间等。Filesystem 可以提供这些详细的信息。
* **访问控制以限制对指定目录的操作：**
* 这是 Filesystem 安全性的核心功能。通过配置文件，用户可以指定哪些目录是“允许的”，而哪些目录是“禁止的”。这可以防止未经授权的访问和操作。

**配置方法**

在 Cherry Studio 的 MCP 服务器设置中，搜索 @modelcontextprotocol/server-filesystem，然后在参数下面添加你打算操作的目录地址。



查找服务器



filesystem

**使用效果**

使用 filesystem，我们可以轻松地操作文件内容，例如新建文件夹、重命名文件夹等。再次提醒，简单的文件操作，免费的 7 B 模型也能做到。



效果

提示词：写一首春天的诗歌， 利用filesystem在桌面（桌面路径C:\Users\chenw\Desktop）新建一个诗歌.txt， 将诗歌内容写入诗歌.txt中。



新建文件

5. **教学案例（待补充）**