Chapter3

MCP Server & Prompt Engineering

主要内容:

- MCP & MCP Server?
- 基本MCP Server配置

- Prompt?
- Prompt Engineering
- 系统提示词

MCP？

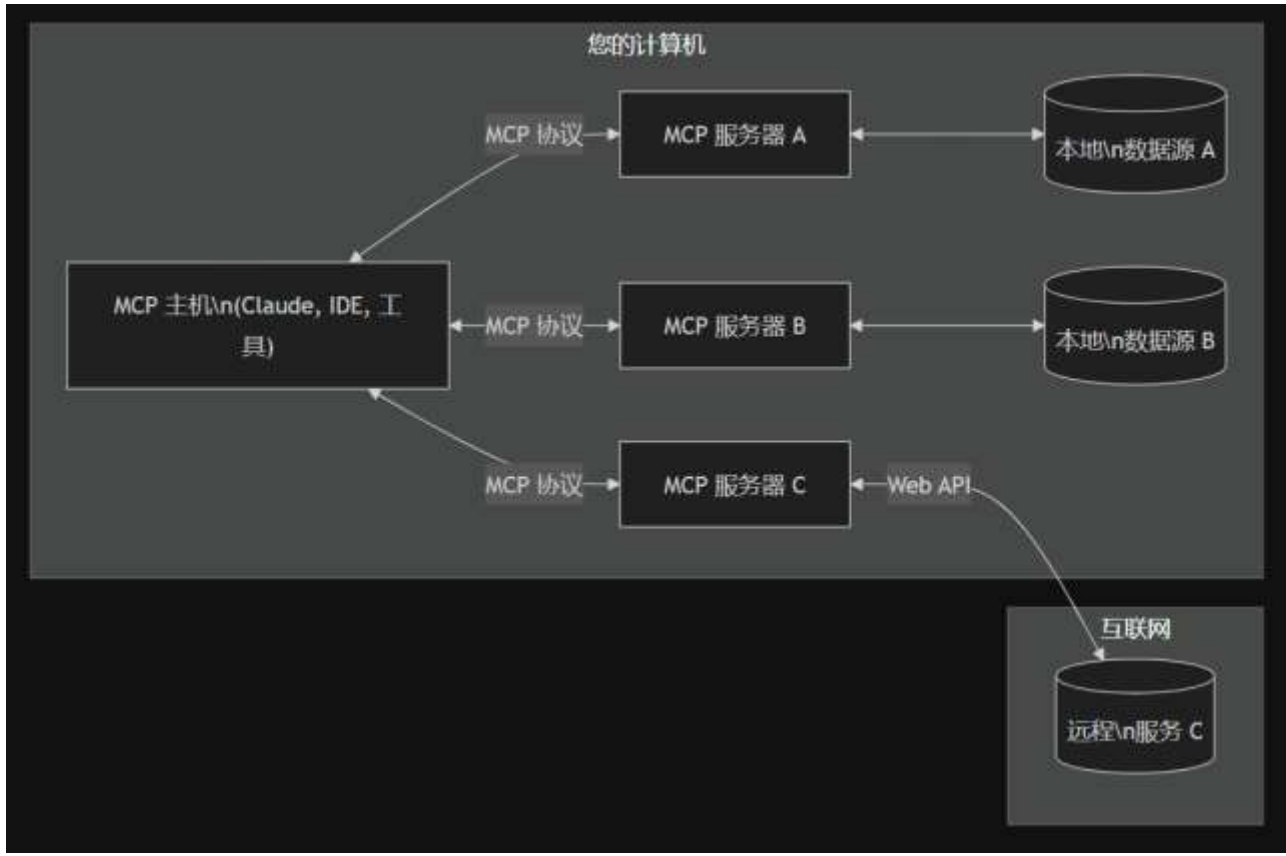
- “NxM”问题：开发人员需要为每个 AI 模型（N）的每个工具（M）编写单独的集成代码。
- 随着模型和工具的成倍增加， 这种方法造成了集成工作的指数级增长。
- 2024.11.24, Anthropic发布MCP
- MCP: Model Context Protocol, 模型上下文协议
- 核心在于提供一个统一的框架，一种开放标准，规范大型语言模型（LLM）与外部数据源及工具之间的交互方式。通过结构化的上下文管理和工具集成，提升模型的功能性、扩展性和协作能力。

MCP

- 试想，平时我想要让大模型去读取我本地的文件、写代码或者查询我在GitHub仓库存放的代码，都需要手动提交文件、截屏或者复制。
- MCP可以让大模型“自己”读取数据源！

MCP Server

- MCP仍然遵循 客户端-服务器架构
- MCP Server一方面通过MCP协议与MCP主机（如IDE、Claude等）进行交互；
- 另一方面通过自身代码接口（filesystem、WebAPI等）访问数据源

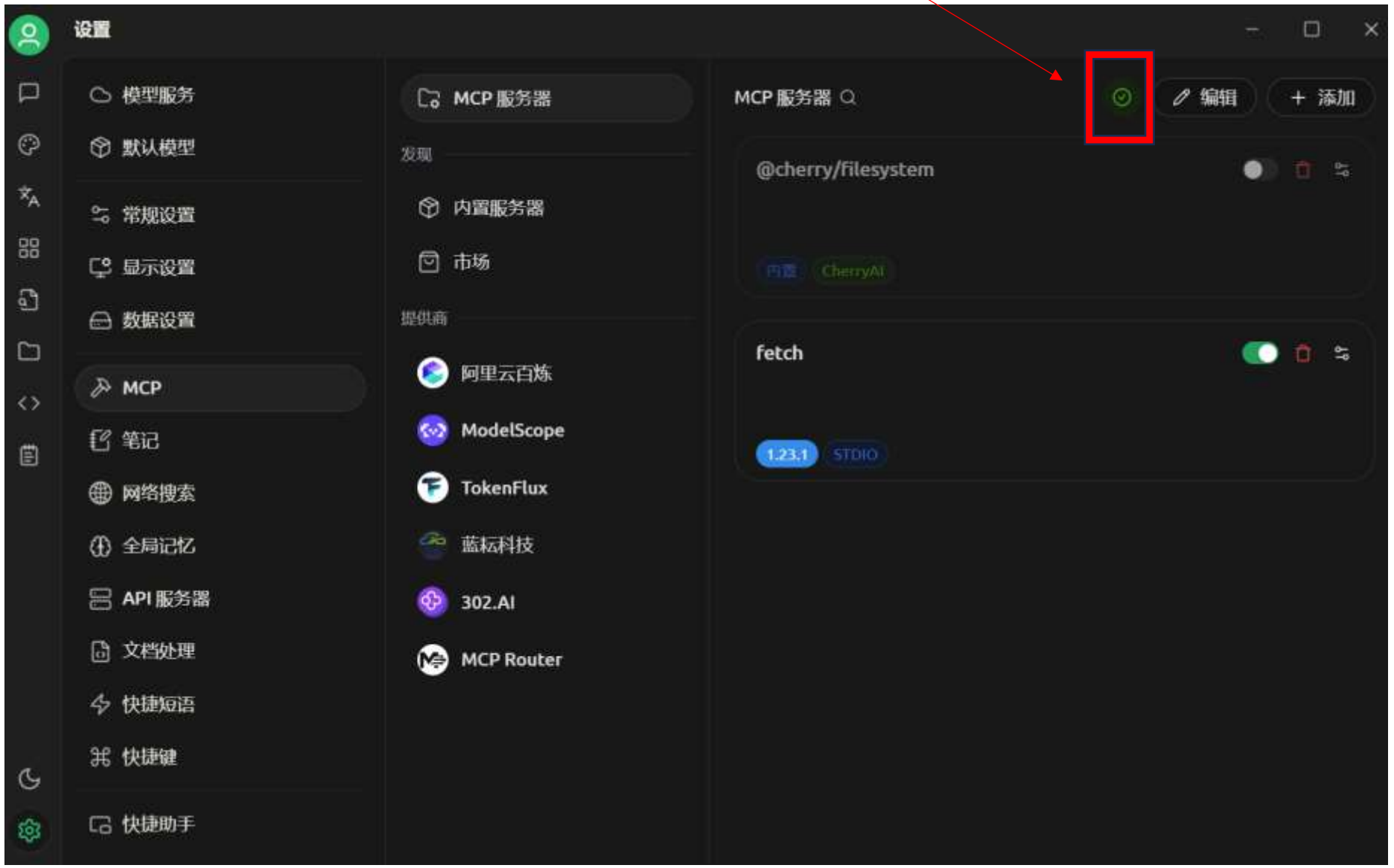


MCP Server

- 从本质上来说，MCP Server是一个可以运行在电脑上的Nodejs | Python程序，如果需要配置相应服务器需要有本地的Python或Node.js环境
- 当前支持 MCP Server 的客户端也不多，例如Cursor、Cline、Wind Surf、Claude App等。
- 但是，好消息是我们的Cherry Studio从较新版本开始支持自动化MCP配置，而且不需要运行环境（一站式服务）
- 于是，我们将以在 Cherry Studio 中配置一个可以访问网页的 MCP Server: Fetch 为例。

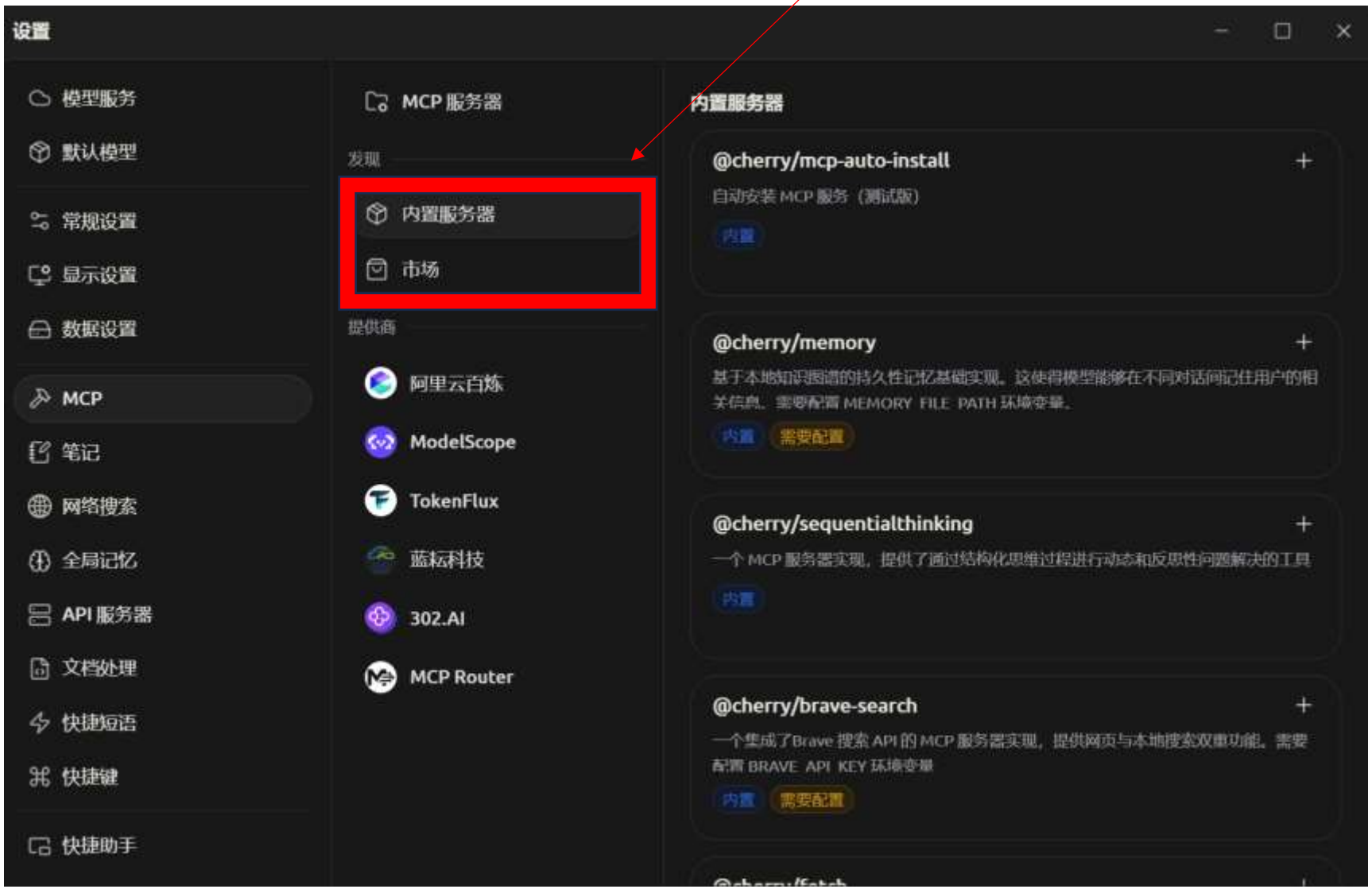
配置

- 进入设置，进入“MCP服务器”选项，右上角会提示环境缺失，我们只需要按照提示安装 **UV** 和 **Bun** 即可；



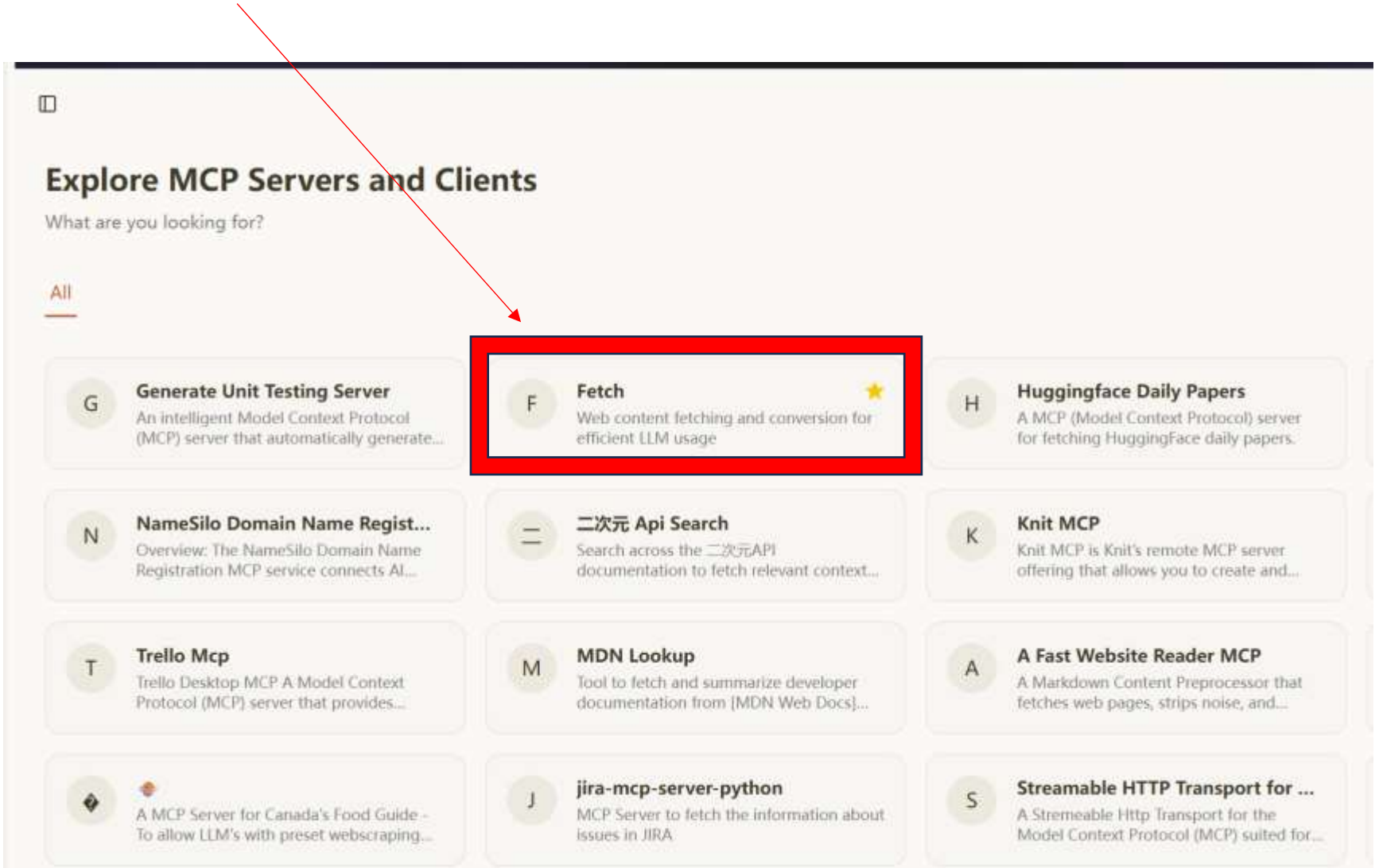
配置

- Cherry Studio内置了一部分服务器，但是显然不能满足我们的需求，接下来我们配置一个可以访问网页内容的MCP服务器：



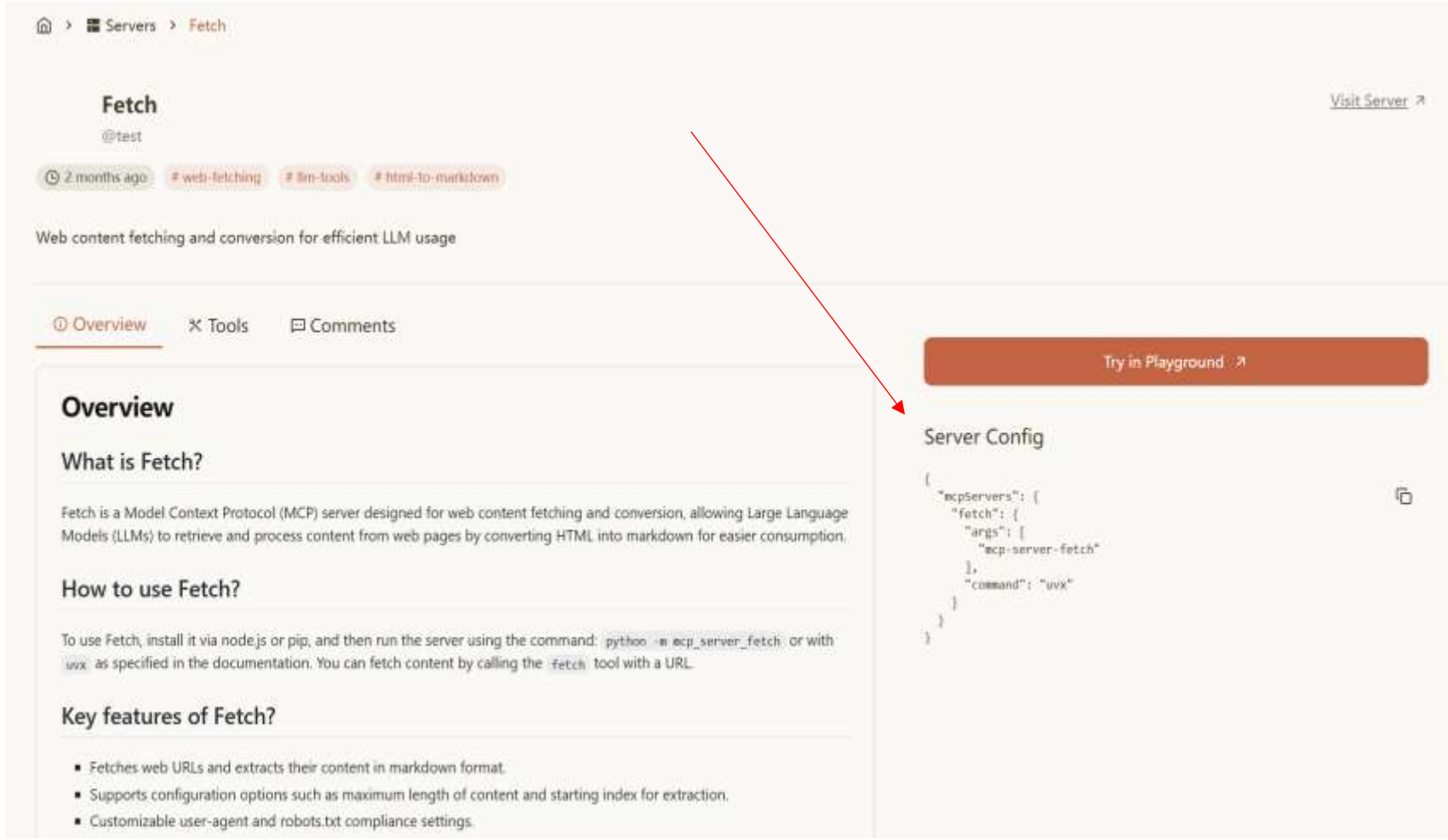
配置

- 点击“市场”，可以访问很多MCP Market，以比较常用的mcp.so为例，我们查找名为“**Fetch**”的MCP服务器：



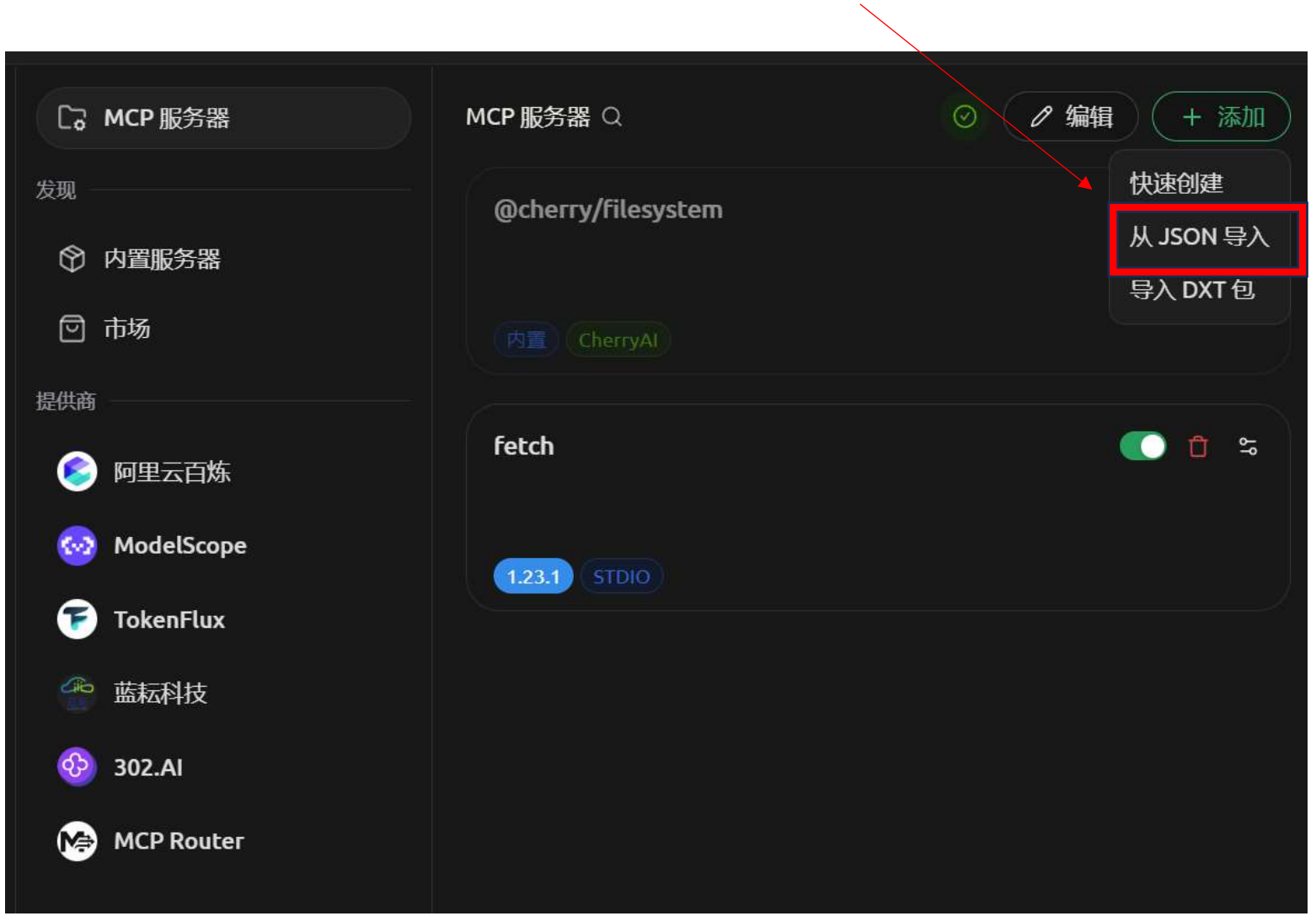
配置

- 将右侧的JSON复制到剪切板
- （由于这个网页实在太慢了，讲义里面有我复制下来的JSON内容，可以直接复制）



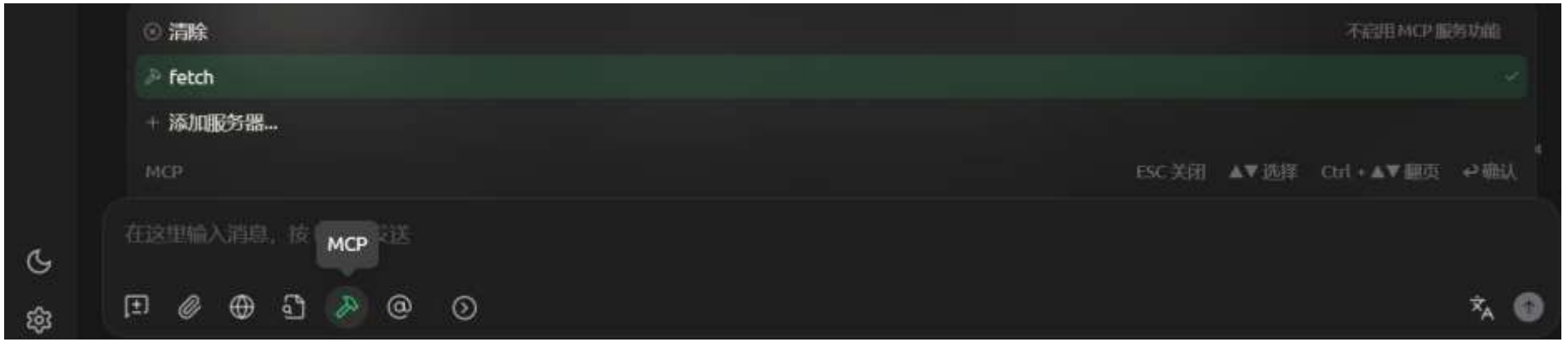
配置

- 回到 Cherry Studio，在“MCP服务器”中选择“添加”→“从JSON导入”→然后直接粘贴即可。
- 如果导入后没有启用，需要重启电脑。



测试

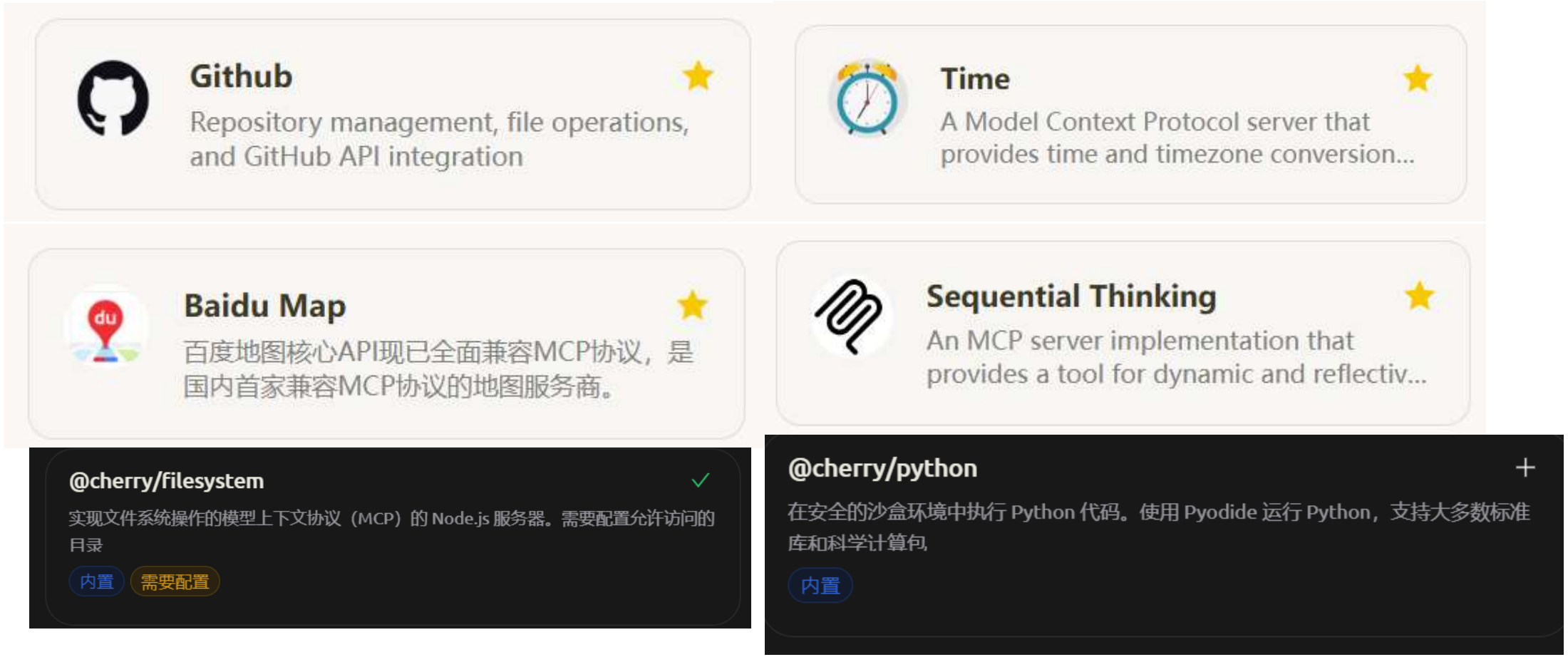
- 成功启用后，就可以将网址交给大模型，让LLM给我们分析/总结网页内容（记得在下方启用MCP Server）。



- 注意：部分网页禁止MCP访问和爬取，例如知乎，所以有时候报错是正常现象。
- 可以用我提供的CSDN网址测试功能。

Other MCPs

- 提供一些有用的MCP服务器（妙妙工具）：



More MCP

- 对于MCP更多的内容，建议大家去阅读MCP中文站的教程，其中有详细的如何设计自己的MCP服务器的教程。
- Cherry Studio是最简单、最实用的配置方式，如果想要在Clien or Cursor里配置还是比较麻烦的（想要额外配置JSON文件和代码运行环境)
- 相关链接已经放在讲义中。

Prompt

《代码有一点bug 你修一修》
《我不要部分代码 你把修改后的完整代码发给我》
《你的代码还是有问题》
《那你重新改，把改好的完整代码给我》
《不是刚刚还能跑 怎么现在跑都跑不了了?》
《你没听懂我的意思吗?不要改变我原来的代码》
《不要给我打印任何调试功能》
《个大我没让你做的 你就别做》
《给我生成完整代码》
《用中文回答我》
《我让你做的功能哪去了》
《听不懂人话是吧》
《不要给我省略代码》
《给我完整的代码》
《我让你写的是完整代码 不是部分函数》
《我的意思是让你改进我的代码，不是重新写一个新的代码》
《你怎么就听不懂啊》
《都说了参考我之前的代码》
《新的代码运行不了 给我回去之前的版本》
《都说了回退版本 你怎么给我写新代码了》
《都说了用C嘎嘎 你怎么给我用Python》
《请不要添加不必要的注释》
《请不要修改我原先的代码基本逻辑》
《帮我修改代码》
《在我的代码上修改...》
《不要改我的变量名!!!》
《你写的代码有问题 跑不了!》
《不要改原有的函数名啊啊啊》
《****你个非物》
《不要污染我的变量》
《不要修改我原来的代码》
《不要添加额外的功能》
《只生成我让你生成的部分》
《不要只生成框架 生成完整代码》

Top 20 AI Prompt Programming Languages

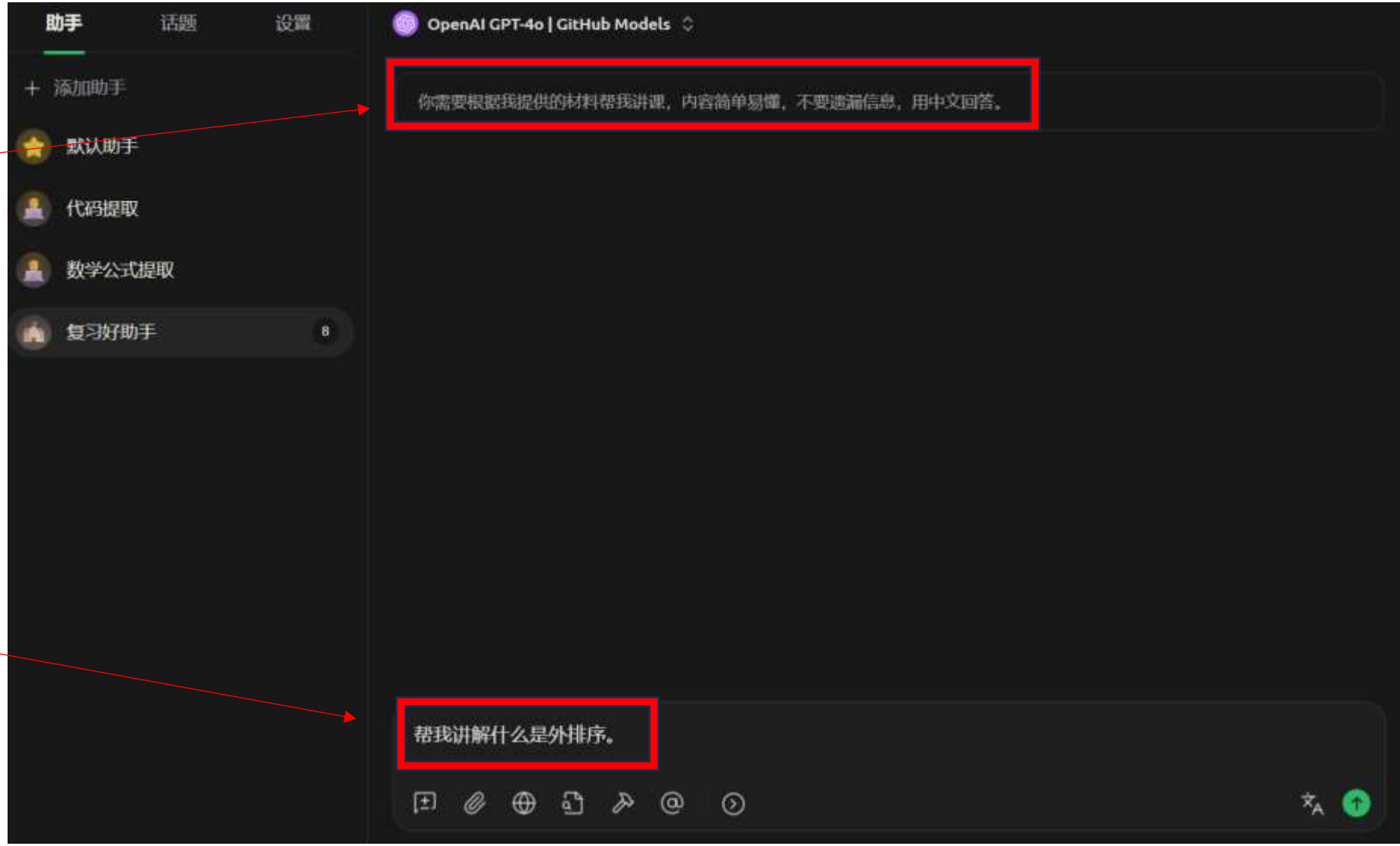
排名	趋势	提示词语言	市场占有率
1	—	给我生成完整可运行的代码	18.762%
2	↑	用中文回答	16.543%
3	↓	我说了要复用现有接口	14.891%
4	↑↑	别又给我生成一堆TODO	12.337%
5	↓	为什么又引入新依赖	10.984%
6	—	这个错误你上次就犯过	9.215%
7	↑	别自作聪明优化	7.638%
8	↓↓	这个API根本不存在	6.492%
9	↑	还是报错	5.871%
10	↑	注释和代码对不上	4.923%
11	↓	就改这里别的别动	4.156%
12	↑↑↑	不要用any啊大哥	3.784%
13	—	能先跑起来再说吗	3.291%
14	↑	你这代码根本编译不过	2.847%
15	↓	按照我给的例子写	2.563%
16	NEW	求你别改我的命名规范	2.194%
17	↓↓	这不是我要的功能	1.928%
18	↑	只要核心逻辑就行	1.675%
19	↓	别一直import你编的库	1.432%
20	NEW	能不能有点记忆力	1.208%

Prompt

- 提示词是我们与大模型交互的媒介，通俗的来讲就是我们的自然语言。
- 严格来说，**提示词工程（Prompt Engineering）**是一门系统的学科，涵盖提示词的设计、优化、上下文管理以及与大模型交互的策略。
- 提示词还分为系统提示词和用户提示词
- 前者是定义AI身份和行为边界的全局指令，后者是用户输入的临时任务指令。前者长期有效，后者仅单次有效。

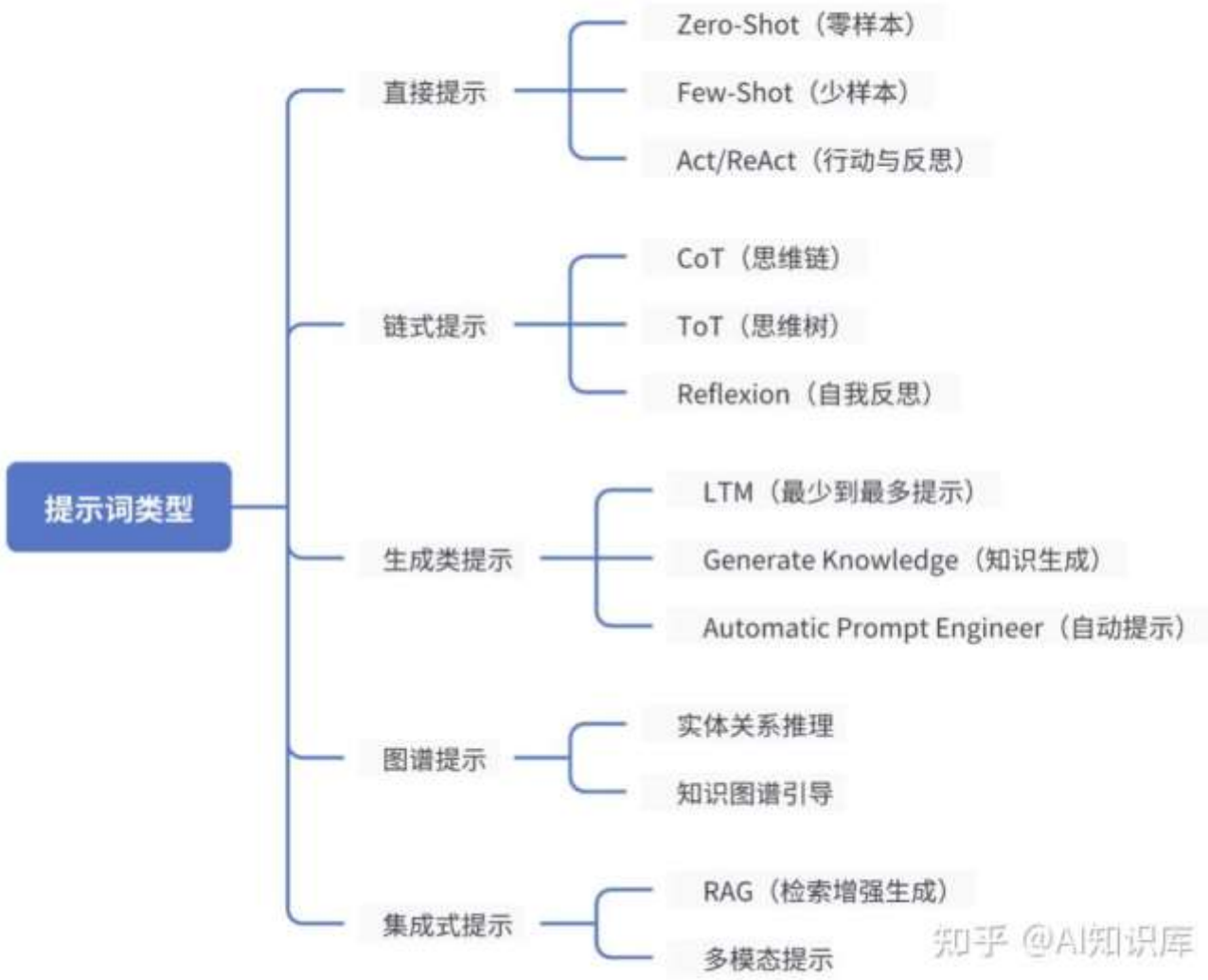
提示词

- 系统提示词
- 用户提示词



Prompt

- Prompt不仅仅是简单地构造输入文本，更是涉及到**对模型行为的深入理解，以及对各种影响因素的综合考量。**通过不断地实验、优化，提示工程师能够设计出最适合特定任务和场景的提示，引导大语言模型生成精准、富有洞察力的输出。



提示词框架

- RACE框架：Role-Action-Context-Expectation
- E.g.:你是《自然》杂志的环境专栏作家，请写一篇800字的深度报道，分析微塑料污染对海洋生态的影响，需引用2024年联合国环境署报告数据，采用’问题-原因-解决方案’结构，语言风格兼顾科学性与可读性。
- RTF框架：Role-Task-Format
- E.g.:你是北京大学信科学院的大一新生，要完成计算概论课程的编程大作业，内容是实现人机交互的亚马逊棋。请完成一份程序功能设计文档，作业使用c++编写，要求面向对象编程，内容简洁。

提示词框架

- Chain-of-Thought：思维链技术，通过引导AI“分步思考”
- 少样本学习：通过提供1-3个示例，让AI快速掌握特定格式。
- 多模态提示：图文联合，通过多种方式向大模型提供信息。

- 总而言之：提示词就是要想大模型提供足够的信息，使之更好地完成给定的任务。

常见误区

- **信息过载与上下文溢出**：大模型通常会有上下文窗口限制，不代表你输入多少字大模型就可以读多少字。
- **模糊指令与主观词汇**：“写一篇有深度的文章”中的”深度”，”生成吸引人的标题”中的”吸引人”，这些词汇对AI而言毫无意义。
- **忽视模型特性差异**：不同AI模型有截然不同的”性格”：文心一言对中文语境理解更准，适合创作类任务；GPT-4在逻辑推理上更强，适合分析类工作；而Claude擅长处理超长文本。
- **缺乏迭代优化意识**：专业提示工程师平均会对一个提示词进行7.3次修改。
- **忽略伦理与安全边界**：

Tendency

- **自动化提示工程APE**：即AI通过用户的简单描述，自动生成包含角色设定、任务分解和输出格式的专业提示词。
- **多模态提示编排**：
- **行业垂直提示词库**：医疗、金融等专业领域正涌现标准化提示词模板库。如美国 Mayo Clinic发布的医疗提示词框架包含：

“患者主诉：[症状]
既往史：[病史]
检查结果：[数据]
请按SOAP格式（主观-客观-评估-计划）生成诊断建议…”

自定义的提示词

- https://github.com/f/awesome-chatgpt-prompts
- 这个GitHub仓库收录了很多有意思的提示词。
- 系统提示词不需要太长，但是作为长期使用的内容，需要让大模型清楚地知道角色定位、用户需求等。
- 例如我比较常用的：
- 代码提取：无论我输入的图片内容是什么，你需要按照原样提取其中的代码，并且指出是什么语言即可。一定要输出代码！
- 数学公式：无论我输入什么内容，你只需要帮我提取其中的数学公式，并返回markdown或LaTeX格式即可。注意保留对齐信息。

自定义提示词

Cherry Studio同样支持自定义提示词，可以获得不同身份和效果的智能体。
左上角“添加助手”选择“默认助手”（也可以选择预设好的智能体，会比自己设计的效果更好）
添加好之后，点击上方的浅灰色框即可自定义提示词（以及其他参数）