微软已发布Playwright Mcp

2025-04-02T16:38:33+08:00

# 前言

MCP协议是由AI公司 Anthropic（claude的母公司）于2024年11月推出的一个针对AI能力的开放协议，旨在标准化大语言模型（LLM）与外部数据源、工具之间的交互接口。其核心设计目标：

* **解耦与标准化**：将LLM与外部系统的集成从定制化开发转变为基于统一协议的通用连接，类似硬件领域的USB-C接口。
* **去中心化**：通过允许开发者自主构建MCP Server，分散AI能力，避免单一厂商垄断知识和服务。
* **安全可控**：通过权限验证和本地化部署，确保数据隐私与操作安全。

而 MCP Server 则是MCP的关键组件，提供如下三种功能：

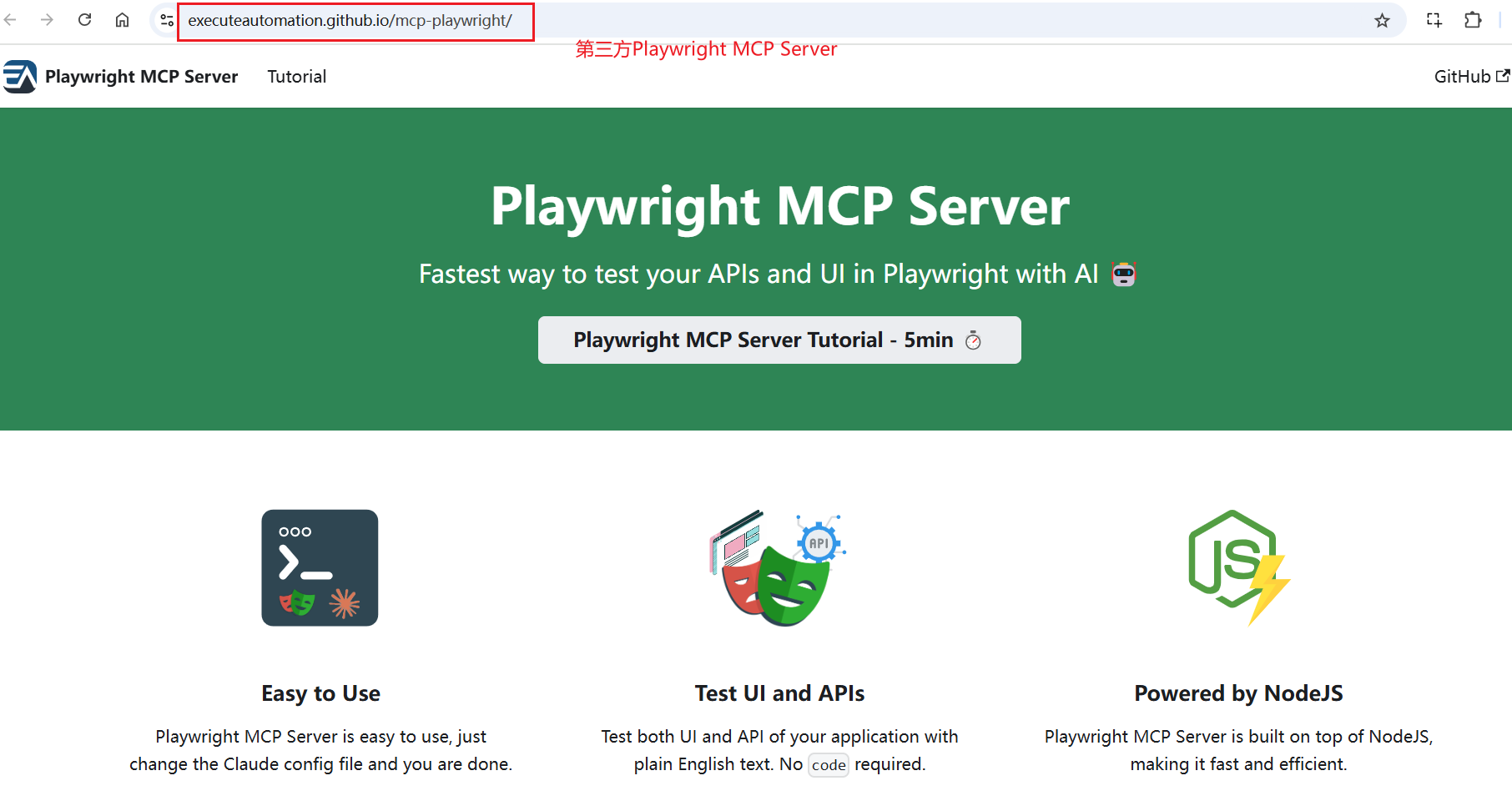
* 资源（Resources）：类似文件的数据，可以被客户端读取，如 API 响应或文件内容。
* 工具（Tools）：可以被 LLM 调用的函数（需要用户批准）。
* 提示（Prompts）：预先编写的模板，帮助用户完成特定任务。

通过这些能力，MCP Server可以为AI应用提供更加丰富的上下文操作私有数据或特定应用的能力，从而增强 LLM 的实用性。

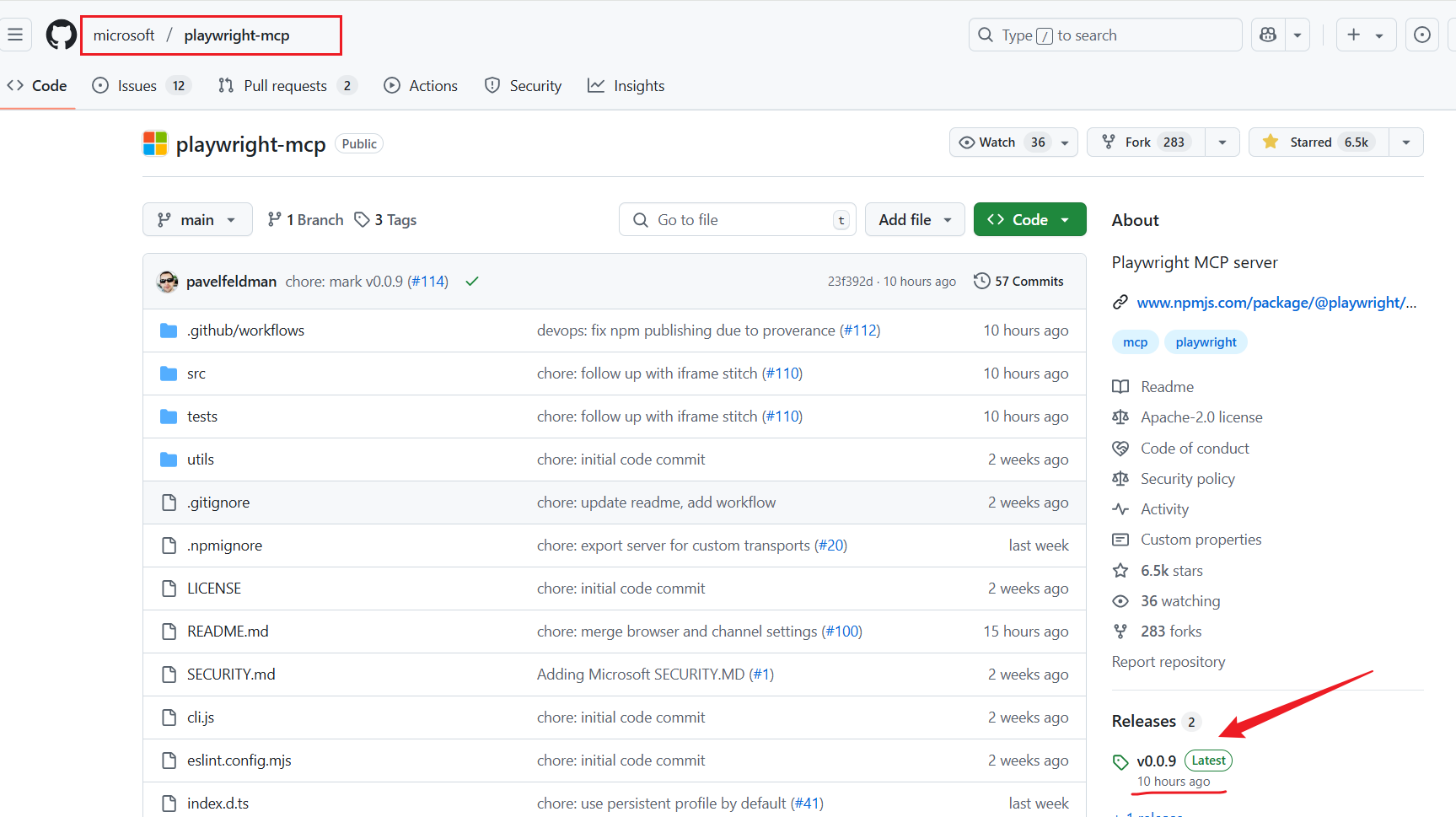
在上一篇文章 【】中，我们已经对 MCP Server有了基本认识，并且提供了一个基本的查询本地桌面的案例，并利用AI编辑器Cursor实现了这个MCP server的调用。

# 微软官方推出Playwright-mcp

MCP Server目前的很多应用中，调度浏览器操作是一个极为高频的场景，基于Playwright这个新锐浏览器工具，也已经有第三方先行推出了MCP Server



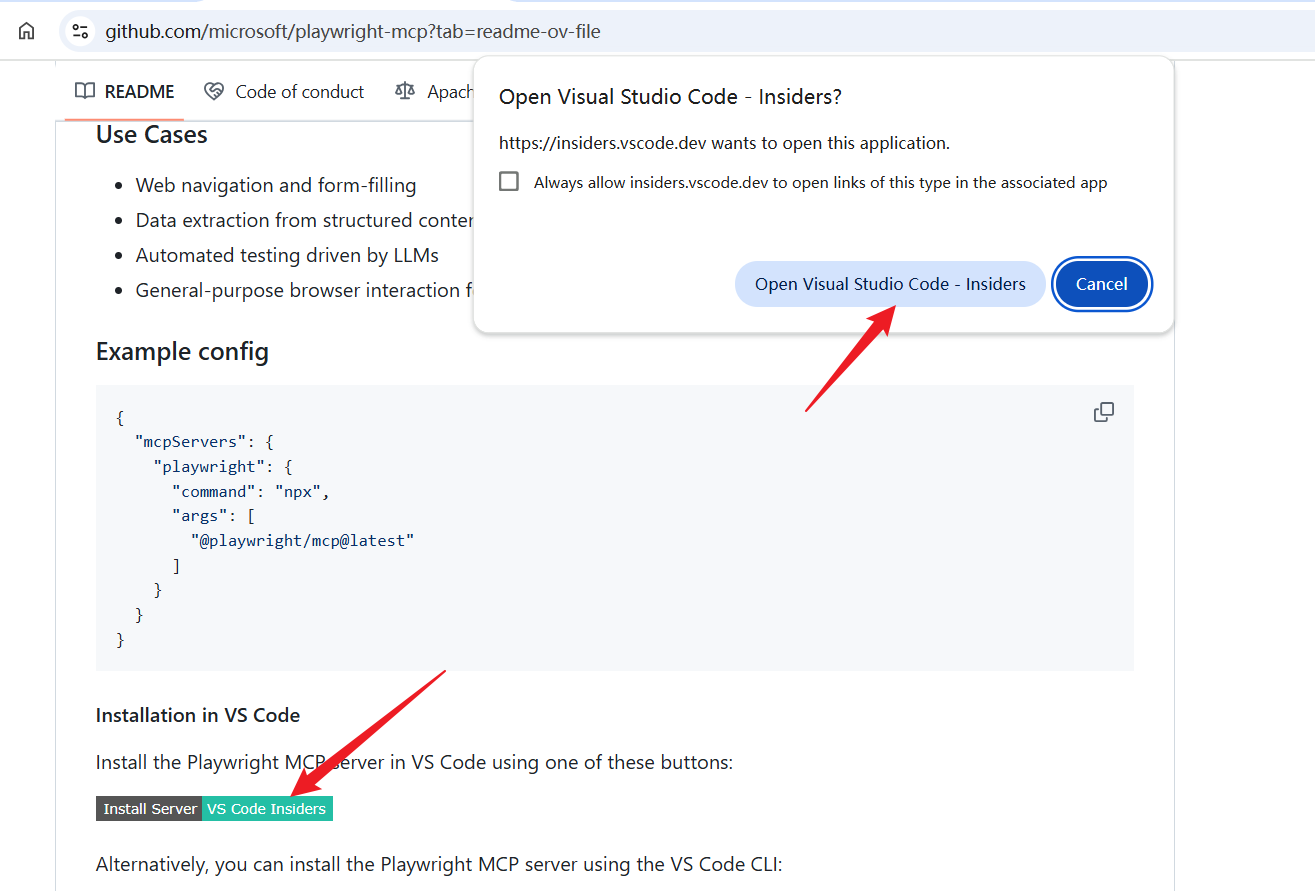
而现在，微软官方也终于推出了Playwright-mcp，可以和Vscode copilot无缝结合使用



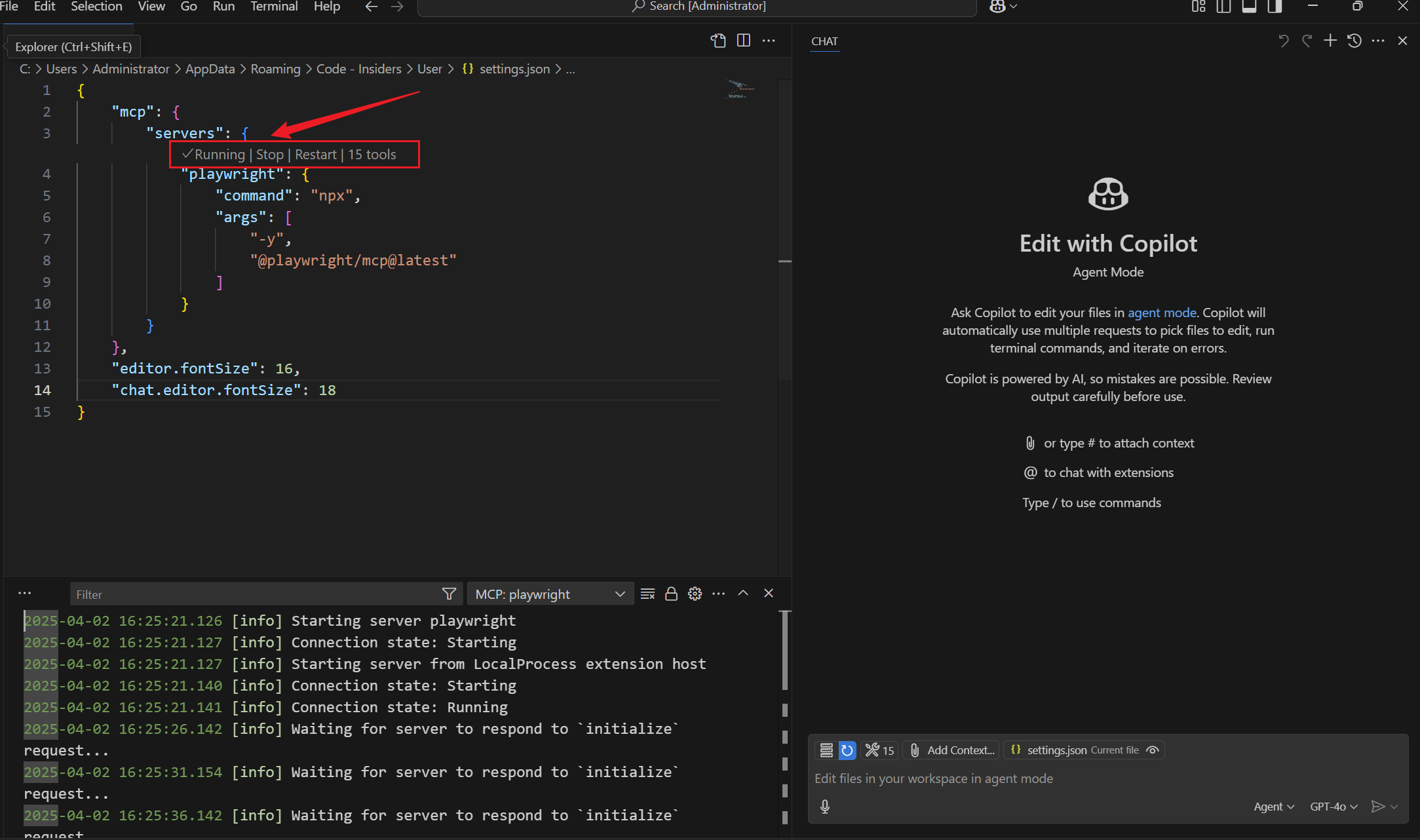
# Playwright-mcp 部署

因为Playwright-mcp刚刚推出，目前 vscode中对它的集成主要还是在vscode的先览版vscode insider中。所以我们可以从vscode官网先下载一个vscode insider 

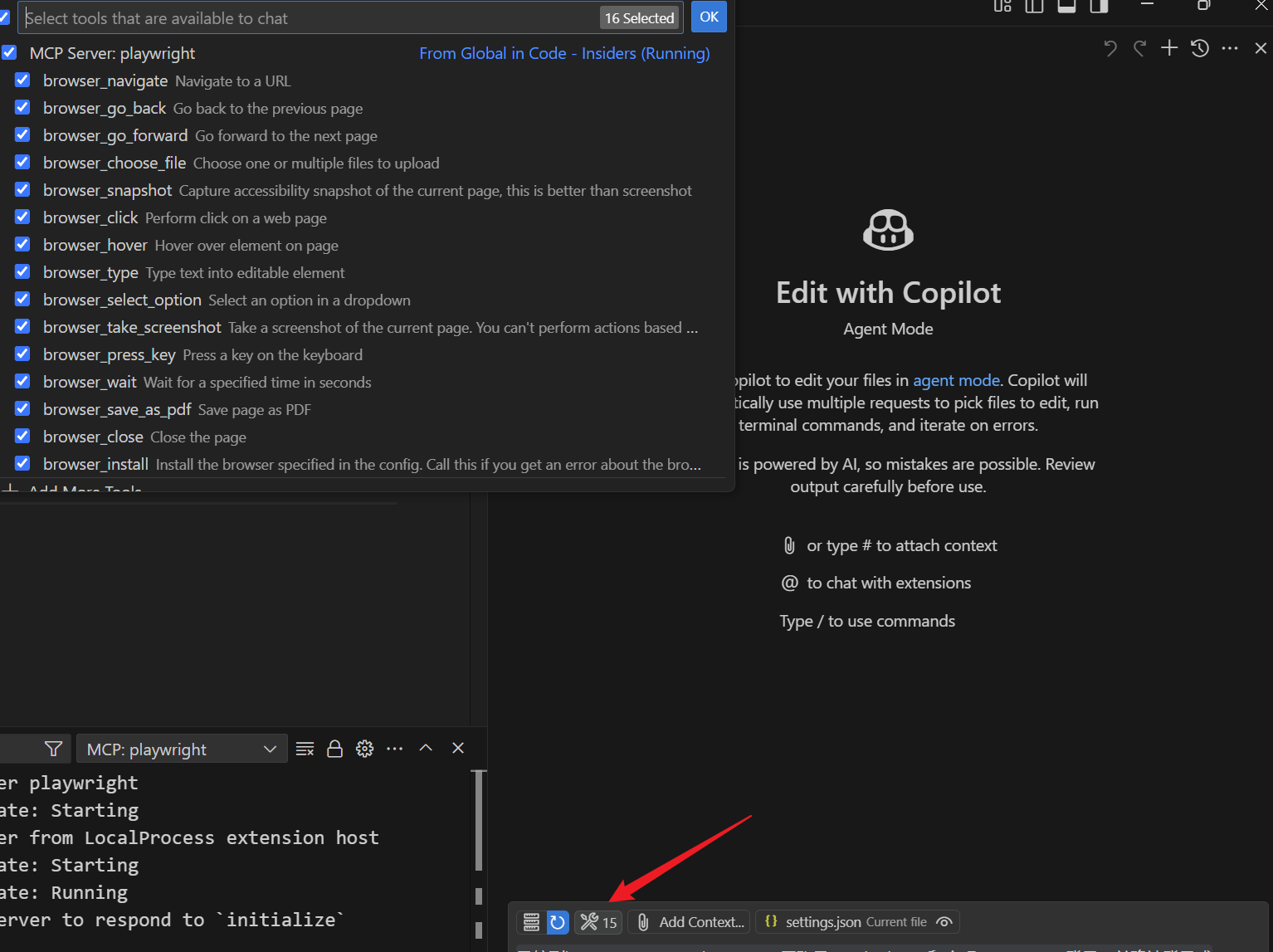
安装好vscode insider后，在github项目主页上，直接点击对应按钮即可进行安装



打开vscode 后，选择 install server，这时会自动配置好mcp server的配置文件，点击start，将mcp server运行起来，这样其实就完成了部署



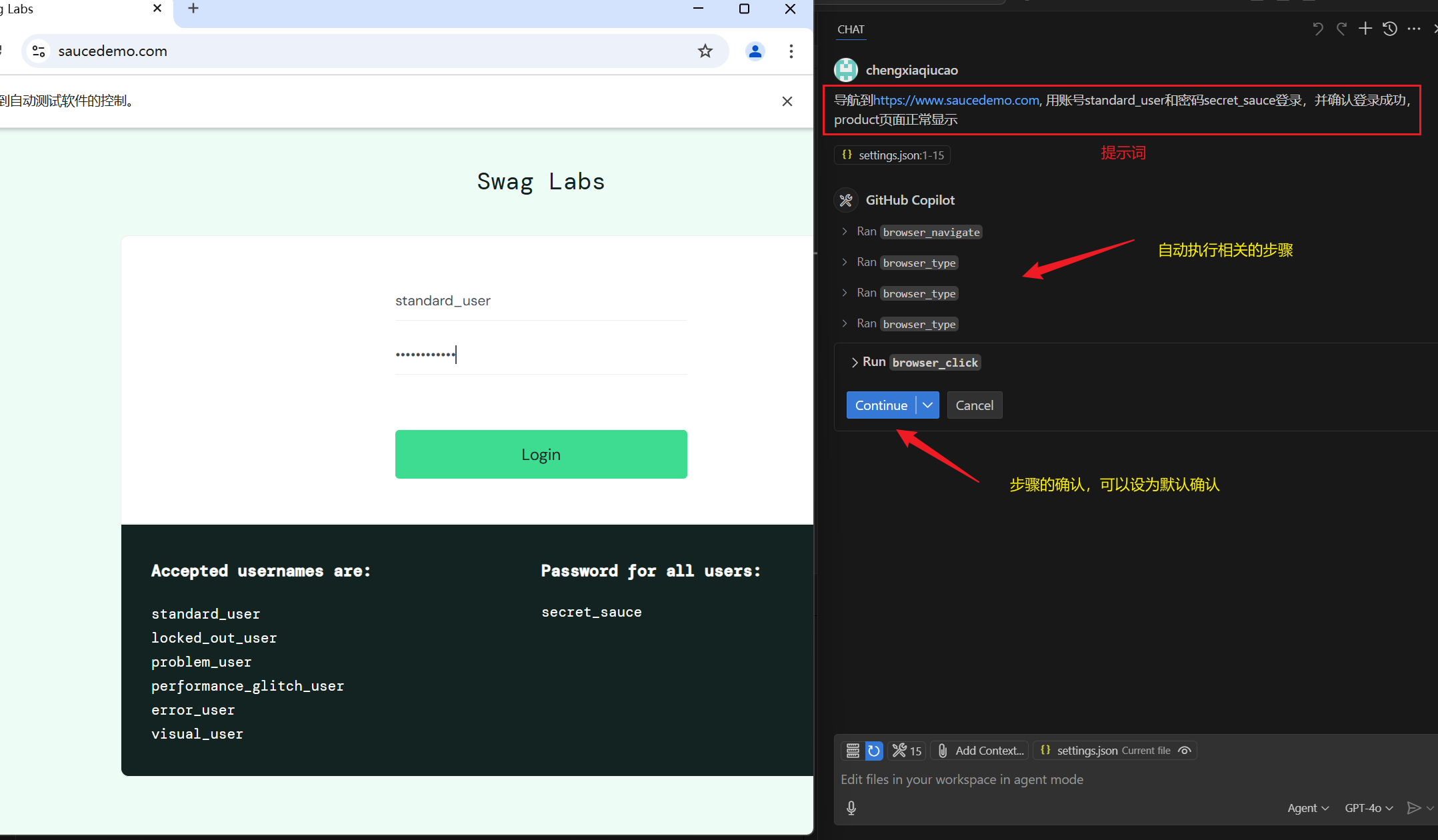
点击copilot对话框中，显示的tools按钮，可以看到目前集成的tools，主要对应playwright的15种主要操作行为

 # 用playwright-mcp无代码完成浏览器操作和验证

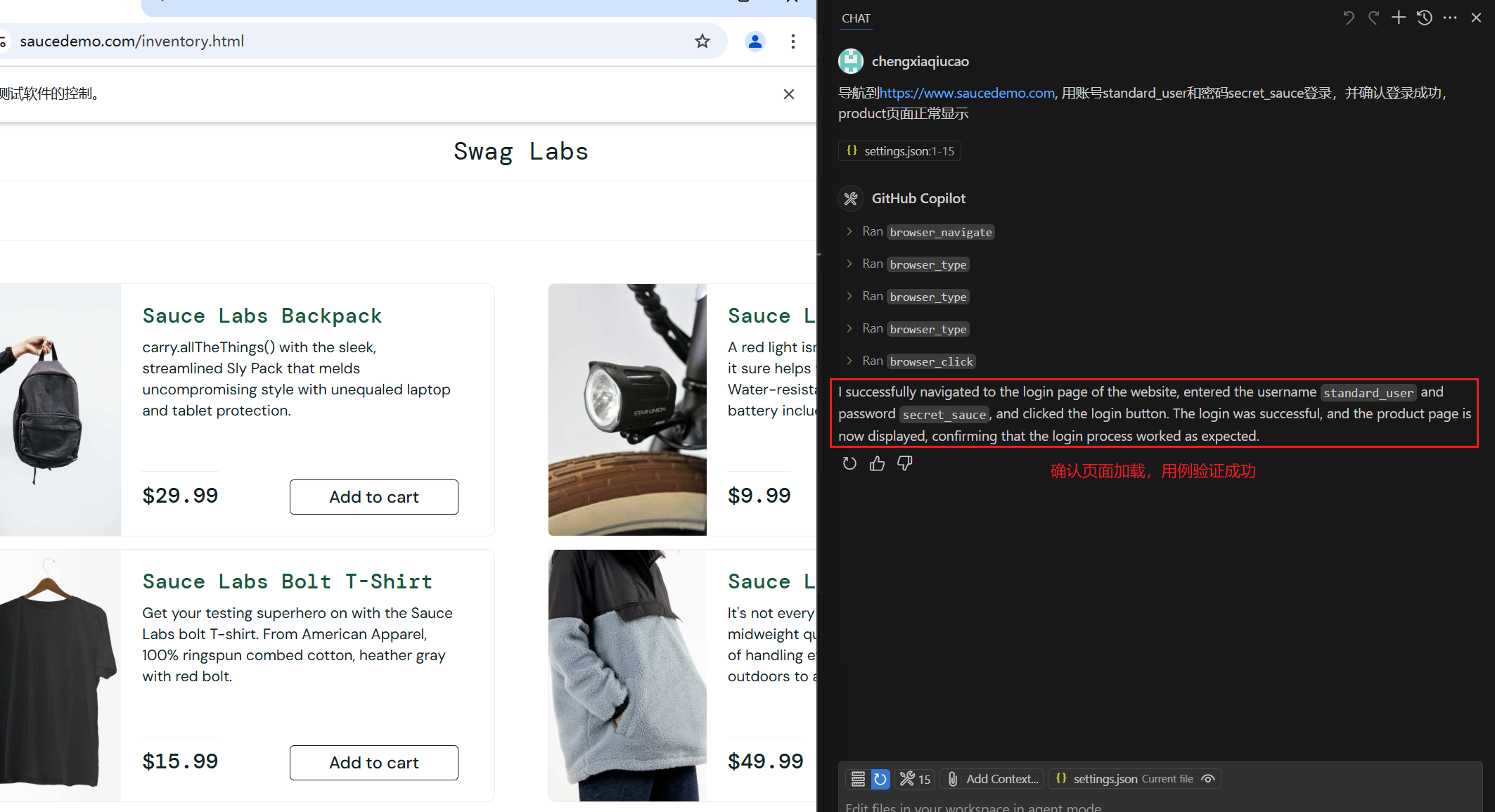
下面我们就看看效果吧。

打开copilot，输入指令：

导航到[https://www.saucedemo.com](vscode-file://vscode-app/c:/Users/Administrator/AppData/Local/Programs/Microsoft%20VS%20Code%20Insiders/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html), 用账号standard\_user和密码secret\_sauce登录，并确认登录成功，product页面正常显示



继续执行，得出结果，用例执行成功



可以看到，这里我们没有编写任何代码，就利用Playwright的mcp完成了对登录用例的执行。

当然实际工作中，通过mcp server的集成，更多是为了我们方便地调用浏览器完成任务执行提供了更多便利！

基于MCP-Server，相信还有更多、更丰富的实践可以帮助到我们！

上一篇文章中，我们介绍了通过vscode insiders 集成微软官方的playwright mcp并执行测试的一个基本效果。

有小伙伴提问是否能在 Cursor 中使用？这里我也进行了尝试，答案是可以成功集成，但有点小坑。 这里就跟大家分享下配置过程，以及Cursor中集成mcp以后的强大能力。让 AI 帮助我们完成大部分基本的功能测试已经不再遥远

## 在Cursor中启用playwright-mcp

### 预置安装

首先，当然要先安装mcp server以及对应的依赖，这里mcp和playwright都是基于nodejs的。所以通过下面命令先安装

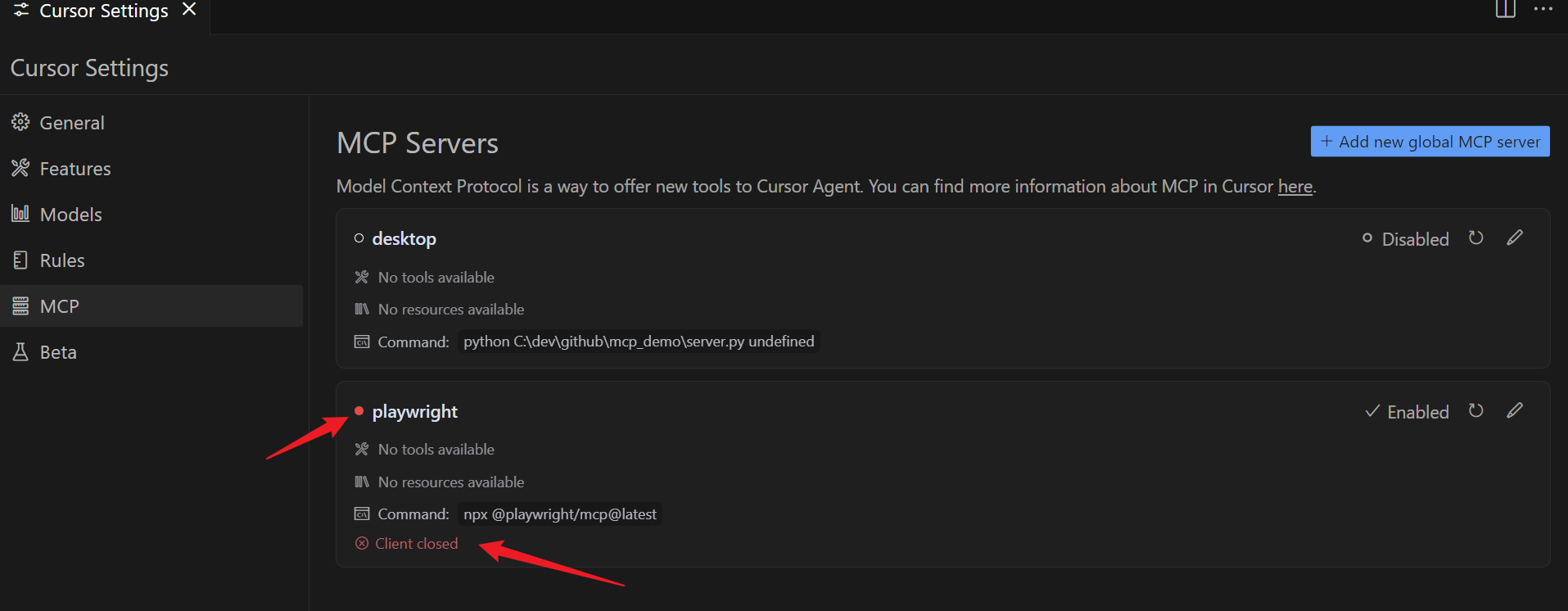
npm install -g @playwright/mcp@latest  
npm install -g @playwright@latest  
npx playwright install --with-deps

### Cursor配置

完成安装后，进入 Cursor 的设置页面，在其MCP部分，添加 MCP server配置，参照Playwright官网的介绍， 添加mcp sever配置：

{  
 "mcpServers": {  
 "playwright": {  
 "command": "npx",  
 "args": [  
 "@playwright/mcp@latest"  
 ]  
 }  
 }  
}

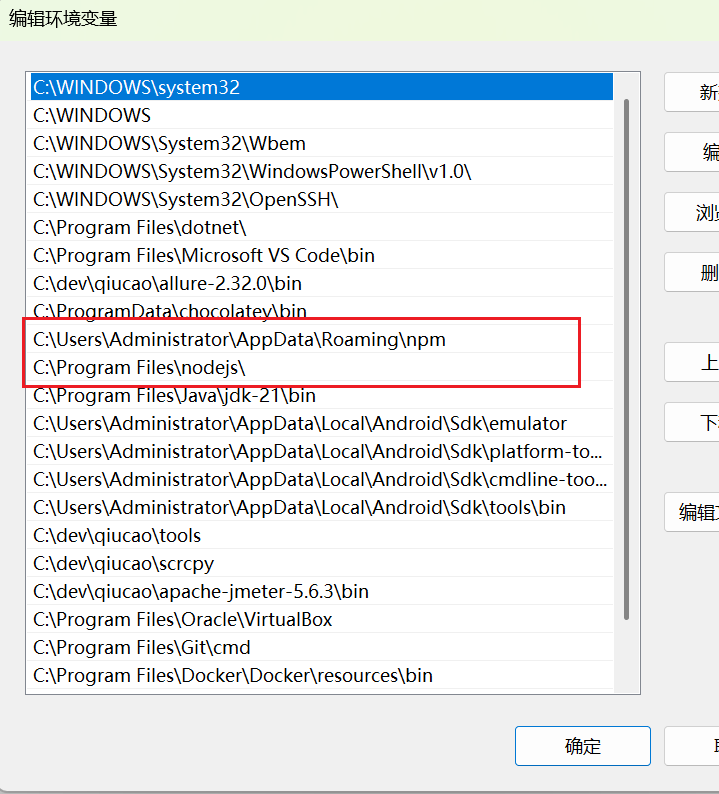
但这时，cursor中的mcp server实际并不能正常运行，会显示 Clint closed



### windows系统安装，要规避两个坑

**第一个坑：npx 包加载路径应该加入搜索路径**

也就是要将全局安装的相关包，放到nodejs的搜索路径中。做法是在windows系统的环境变量中，把npm包目录加入path中，并放到nodejs之前



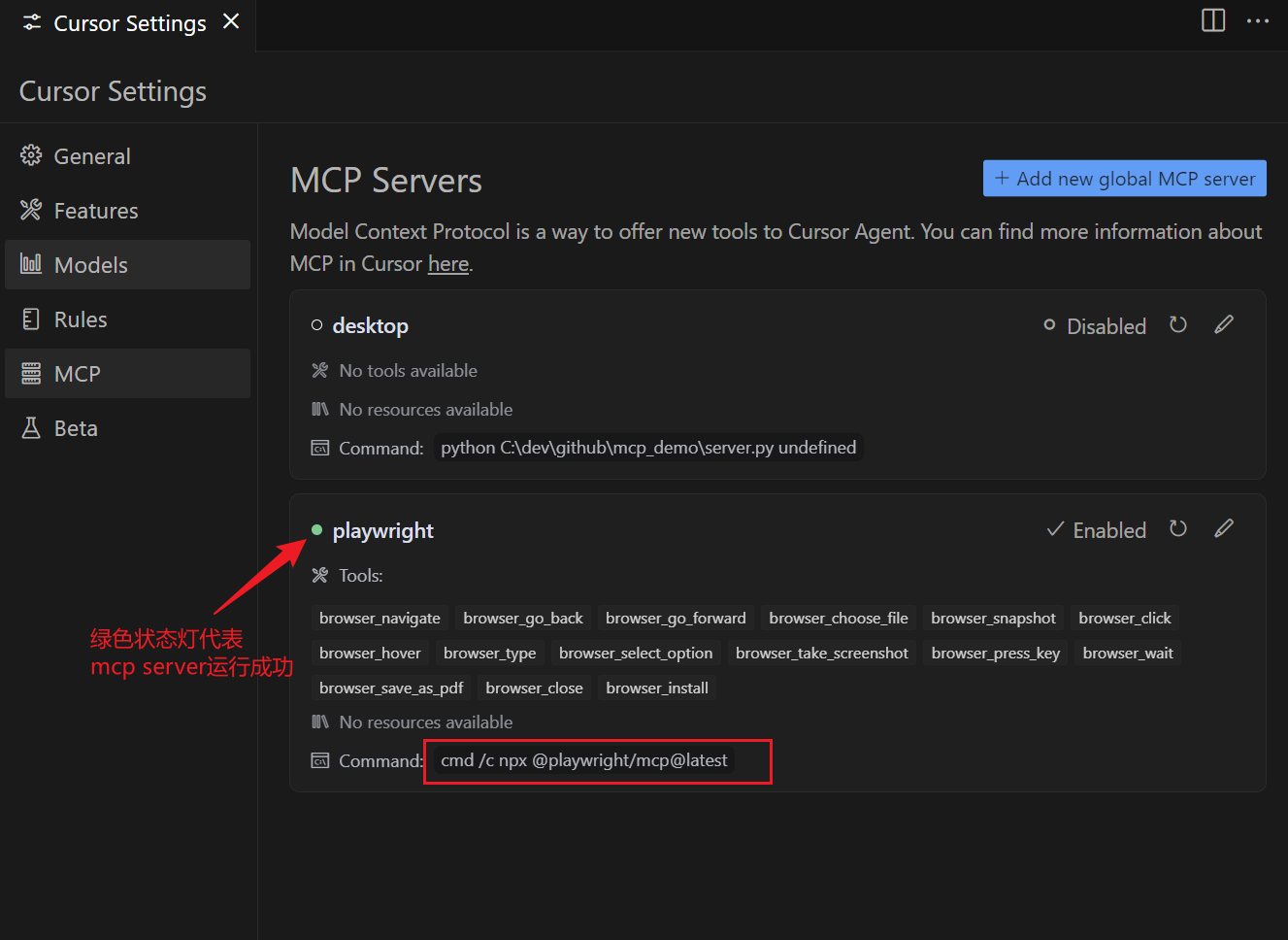
**第二个坑：修改mcp命令行调用命令**

cursor中调用npx，其实是通过命令行调用的，但windows系统下，直接通过命令行调用npx，cursor本身并不能拿到调用状态，所以这里要用 cmd /c 命令调用, 获取命令句柄

也就是上面的配置文件，修改成：

{  
 "mcpServers": {  
 "playwright": {  
 "command": "cmd /c npx",  
 "args": [  
 "@playwright/mcp@latest"  
 ]  
 }  
 }  
}

修改以上配置后，重新加载cursor，这时会启动一个命令行窗口，其实就是playwright-mcp，不要关闭，然后可以看到 Cursor中mcp也正常启动了



## 让 playwright-mcp 代替我们执行测试

我们在Cursor中输入一个基本的任务指令：

访问www.saucedemo.com网站，使用problem\_user登录，验证该网站完整流程并记录发现的问题

以下是完整执行过程

我将帮您访问 Sauce Demo 网站并使用 problem\_user 账号进行完整测试流程验证。让我们开始吧。  
  
首先，让我访问网站：  
  
Called MCP toolbrowser\_navigate  
  
让我使用 problem\_user 账号登录：  
  
Called MCP toolbrowser\_type  
  
现在输入密码：  
  
Called MCP toolbrowser\_type  
  
点击登录按钮：  
  
Called MCP toolbrowser\_click  
  
让我们开始测试并记录问题。首先我注意到以下问题：  
  
1. 商品图片问题：  
  
所有商品的图片都显示相同的图片（狗的图片），这明显是一个bug，因为每个商品应该显示其独特的产品图片。  
  
让我尝试添加商品到购物车：  
  
Called MCP toolbrowser\_click  
  
让我点击购物车图标查看购物车：  
  
Called MCP toolbrowser\_click  
  
让我们返回商品列表并尝试使用排序功能：  
  
Called MCP toolbrowser\_click  
  
让我尝试使用排序功能：  
  
Called MCP toolbrowser\_click  
  
Called MCP toolbrowser\_select\_option  
  
基于测试过程，我发现了以下问题：  
  
1. 图片显示问题：  
  
- 所有商品的图片都显示为相同的狗图片，而不是各自的产品图片  
  
- 这影响了用户对产品的视觉识别  
  
1. 购物车功能问题：  
  
- 点击商品图片或标题时，没有正确跳转到商品详情页，而是直接跳转到了一个固定的商品页面  
  
- 这影响了用户浏览和选择商品的体验  
  
1. 排序功能问题：  
  
- 选择不同的排序选项（如按价格从低到高）后，商品列表顺序没有发生变化  
  
- 排序功能完全失效  
  
1. 导航问题：  
  
- 点击商品时总是跳转到同一个商品页面，而不是所点击的具体商品  
  
- 这严重影响了用户的购物体验  
  
1. 按钮状态问题：  
  
- 某些商品的"Add to cart"和"Remove"按钮状态显示不一致  
  
- 这可能导致用户对购物车中的商品状态产生混淆  
  
这些问题严重影响了网站的基本功能和用户体验。problem\_user 账号确实展示了多个需要修复的问题，这些问题涵盖了UI显示、功能操作和用户体验等多个方面。  
  
是否需要我继续测试其他功能，比如结账流程？  
  
请继续测试  
  
好的，我将继续测试结账流程。首先让我们添加一些商品到购物车，然后进行结账：  
  
Called MCP toolbrowser\_click  
  
现在让我们点击购物车图标进入购物车页面：  
  
Called MCP toolbrowser\_click  
  
我发现点击购物车图标又出现了一个问题 - 它把我们带到了商品详情页而不是购物车页面。让我们尝试点击页面右上角的购物车图标：  
  
Called MCP toolbrowser\_navigate  
  
让我们继续结账流程，点击Checkout按钮：  
  
Called MCP toolbrowser\_click  
  
让我们填写个人信息：  
  
Called MCP toolbrowser\_type  
  
Called MCP toolbrowser\_type  
  
Called MCP toolbrowser\_type  
  
点击Continue继续：  
  
Called MCP toolbrowser\_click  
  
我发现了一个新的问题：虽然我们输入了姓氏，但系统显示"Last Name is required"的错误。这说明在输入字段上存在问题。让我们再次尝试输入姓氏：  
  
Called MCP toolbrowser\_type  
  
基于测试过程，我发现了以下新的问题：  
  
1. 购物车导航问题：  
  
- 点击商品列表页面的购物车图标会错误地导航到商品详情页  
  
- 必须手动输入购物车URL才能访问购物车页面  
  
1. 结账表单问题：  
  
- 姓氏（Last Name）输入框存在严重问题  
  
- 即使输入了姓氏，系统仍然显示"Last Name is required"错误  
  
- 这个问题阻止了用户完成结账流程  
  
1. 购物车商品显示问题：  
  
- 购物车中显示的商品数量与实际添加的商品数量不一致  
  
- 一些未添加的商品也出现在购物车中  
  
这些问题与之前发现的问题一起，严重影响了整个购物流程：  
  
- 用户无法正常浏览商品（图片问题）  
  
- 无法正确添加商品到购物车（购物车功能问题）  
  
- 无法使用排序功能（排序失效）  
  
- 无法正常访问购物车（导航问题）  
  
- 无法完成结账流程（表单验证问题）  
  
这些问题使得 problem\_user 账号几乎无法完成正常的购物流程。每个主要功能都存在不同程度的问题，这可能是为了测试目的而特意设置的。  
  
是否需要我继续测试其他功能，比如菜单栏的功能？  
  
完成测试，请退出登录并汇总发现的问题  
  
好的，让我们先点击菜单按钮然后退出登录：  
  
Called MCP toolbrowser\_click  
  
Called MCP toolbrowser\_click  
  
现在我来汇总使用 problem\_user 账号测试发现的所有问题：  
  
### 1. 商品展示问题  
  
- 所有商品图片显示为相同的狗图片，而不是各自的产品图片  
  
- 这严重影响了用户对产品的视觉识别和购物体验  
  
### 2. 导航问题  
  
- 点击商品图片或标题时，总是跳转到固定的商品页面，而不是所点击的具体商品  
  
- 点击购物车图标错误地导航到商品详情页，而不是购物车页面  
  
- 必须手动输入购物车URL才能正确访问购物车页面  
  
### 3. 商品排序功能问题  
  
- 选择不同的排序选项（如按价格从低到高）后，商品列表顺序没有任何变化  
  
- 排序功能完全失效，无法帮助用户按需要的方式浏览商品  
  
### 4. 购物车功能问题  
  
- 购物车中显示的商品与实际添加的商品不一致  
  
- 一些未添加的商品也出现在购物车中  
  
- "Add to cart"和"Remove"按钮状态显示不一致，造成用户混淆  
  
### 5. 结账流程问题  
  
- 姓氏（Last Name）输入框存在严重问题  
  
- 即使输入了姓氏，系统仍然显示"Last Name is required"错误  
  
- 这个验证问题完全阻止了用户完成结账流程  
  
### 6. 用户体验影响  
  
- 以上问题的综合效果导致：  
  
- 用户无法正确浏览和选择商品  
  
- 无法管理购物车内容  
  
- 无法完成购物流程  
  
- 整体购物体验严重受损  
  
### 7. 功能可用性评估  
  
- 基本功能几乎都存在问题  
  
- 用户无法完成正常的购物流程  
  
- 这个账号可能是专门用于测试目的，展示了各种可能的问题场景  
  
这些问题涵盖了电子商务网站的核心功能，包括商品展示、导航、购物车管理和结账流程。每个环节都存在不同程度的问题，使得整个购物体验变得极其困难和令人沮丧。这些问题的存在可能是为了测试目的，帮助开发人员和测试人员理解各种可能的错误场景。

可以看到，通过 Cursor + Playwright-mcp ，AI 已经可以通过我们交代的基本指令，主动帮助我们去覆盖主要的功能测试场景并发现和记录问题。假以时日，完全可以很大程度上替代掉比较基础功能测试任务，而测试人员则可以更关注在高价值的深度问题挖掘上。当然，另一方面，基础功能测试人员，确实也要有相当的危机意识，技能提升和扩展边界刻不容缓。



欢迎关注秋草的测试技能体系化进阶提升课程，公众号回复 大纲 了解课程详细目录