前端智能化实战-WebMCP-WebSkills-WebAgent-02_原文

2026年06月14日 14:21

00:09

各位屏幕前的开发者朋友们，大家晚上好。欢迎来到由open tiny开源社区和git code联合举办的线上技术直播。我是今天的主讲人k go。首先非常感谢大家在忙碌的工作和学习中抽出时间来参加咱们这次直播。无论是刚进入直播间的同学，还是提前预约守候的老朋友，都先给大家道一声感谢，也麻烦大家动动手给直播间点个赞，加个关注，后续我们还会持续分享更多实用的技术干货。

00:40

今天我将和大家探讨一个非常前沿且重要的一个话题，就从AI agent到web MCP，我们一起来看一看智能体在与外部应用的连接方式正在发生怎样的一个深刻的变革。这不仅仅是技术的演进，更是一场关于AI如何更更高效更可靠的与我们日常使用的外部世界进行交互的革命。我们一步一步把web MCP这个前沿的技术进行拆解和实战，把这些技术讲透练会，确保大家听完今天的直播就能直接把学到的知识点用到自己的项目里。

01:20

简单分正式分享之前，简单跟大家讲一下今天我们直播的流程。首先主要是分成三个部分，首先会给大家讲一下一些背景知识，包括AI agent还有MCP等等。接下来我们会大概用半个小时的左右带着大家一起完成。就是把自己的一个外部应用怎么样改造，通过web MCP改造成一个智能应用。然后这个是一个实操的过程，然后全程会实时的演示代码和调试环境。大家有任何的疑问，也可以在弹幕里面去留言。我们在我会在实操结束后，专门留大概5到10分钟钟的时间集中给大家解答问题。然后另外就是直播过程中也会有一些小互动和福利，我们会针对优秀提问的开发者抽出三位朋友送出我们周边礼品。

02:14

大家全程跟上节奏，不要走开。话不多说，咱们正式开始。大家做好准备，有不清楚的地方随时弹幕告诉我。如果你手上有VS code之类的代码编辑器，也可以提前打开。我们后面会有实操，大家可以跟着我一起做。

02:34

本次分享的话会分成五个部分。首先我们会回顾一下AI agent从单纯的对话能力到具备行动能力的一个进化的一个历程。然后会给大家介绍一下MCD协议它是怎么样为AI agent调用外部工具提供标准化支持的。接着我们会对比当前业界主流的AI操作web应用的方案，包括基于视觉识别的、基于端的、基于无障碍素等等，分析它们的优劣。最后我们将会重点介绍web是MCP这一更优的连接方式，并展望其未来的发展。最后通过我们欧文ti提供的web MCB解决方案，web NCB杠SDK带大家进行实操。

03:21

怎么样通过三个简单的步骤就能把你的web应用变成智能应用，实现被AI操控。在正式开始之前，我们先来思考一个问题，就是为什么让AI去操作这个web页面，它会非常的困难。我们都知道大元模型现在能力已经非常强大了，它能够理解非常复杂的指令，并且生成高质量的文本。但是甚至他在编码，生成PPT等等这些领域其实都已经非常强了。但是为什么他操作跟web应用进行交互的时候，它就显得非常的笨拙、低效，甚至是非常的不可靠呢？这就好比你请了一个天才的助手，结果让他蒙着眼睛去操作电脑，只能靠不停的屏幕截图来看屏幕上有什么，结果就是又慢又贵又脆弱。这背后的核心问题就是我们怎么样让AI能够跨越从理解到执行，特别是操作外部应用的这样一个鸿沟。让AI能够像人类一样灵活高效地操作外部应用，这个就是我们今天要探讨的核心。

04:37

首先让我们进入第一个部分。回顾一下这个AI agent，它怎么从一个只能被动对话的机器人变成进化成能够主动执行任务的智能体的。一个典型的AI agent，传统的大语言模型的话，它其实就像一个知识非常渊博，但是却不会动手的一个顾问。你问他什么他就答什么，他没有办法去自己动手去完成任务。但是AI agent不一样，它是一个拥有大脑并且有手脚和工具，能够执行任务的一个智能体。他能够主动理解你的目标，去规划怎么样去完成这个目标，就是一系列的步骤，然后调用合适的工具，最后再进行执行后还会进行反思和调整。这个就是从被动响应到主动执行的一个根本性的一个转变。

05:36

我们都知道一个典型的AI agent，它通常由四个部分组成，就是感知、规划、执行和反馈。它通过感知模块来理解用户的目标和环境的一些信息。就比如说我们用用比如说VS code copy a或者是tree去帮我们写代码。那么他就能够理解你的需求，并且感知你当前在一个什么样的一个项目里面，你的代码大概是一个什么样的情况，他都能够感知到。再由规划模块将你的这个需求进行一个分解，分解成一个具体一些一系列的具体的步骤。然后再通过调用相应的一个工具和执行来最终执行这些步骤。最后还会去通过反馈和反思来去检查这个结果能不能满足最初的目标，并且做相应的一个调整。这个闭环使得AI agent能够持续地学习和改进。

06:41

讲完了这个MAI agent的基本的一个知识之后，我们来再来看一下AI agent的到底是怎么样跟外部世界交互的这就引出了我们的第二个部分MCB协议。这是一个anthropic这边这家公司设想的一个规范的一个用来规范AI agent调用外部工具的一个这样一个协议。并且它有专门的包括text crept，包括python的一个相应的SDK，我们可以去使用。

07:21

那么这个MCP它到底是个什么东西呢？我们大家都用过手机电脑都知道要要给比如说要给手机电脑充电。我们现在比较流行的就是用type c的一个充电器一个接口，这个MCP就可以理解为AI世界的type c type type c接口。在过去我们要连接不同的设备需要非常多不同的接口。有USB或者是type c，或者是苹果的这种各种非常的繁琐。但是type c的出现统一了这个标准，我们几乎可以一用一根线连接几乎所有的设备。

07:57

同样的MCP协议的目标也是为了让AI应用跟外部工具之间的交互提供一套统一的标准。它是一个开放的协议，旨在简化集成的过程，让任何AI应用都能够方便的去连接到任何符合规范的工具或领域，让整个集成变得非常的高效和简单。这里我自己也用define这个智能体搭建平台做了一个AI agent。我用它去集成了123306官方的MCP工具。这样的话我这个AI卷的就可以去调这个工具去帮我去查火车票。这里由于时间关系不给大家演示，后续我们还会有更多的一个视频的一个演示会给大家去呈现。

08:48

MCP的话它的经典的一个它其实是一个经典的客户端服务器的一个模型。其中MCP host是我们的AI的一个应用。比如说可copilot或者一些AI助手，它都可以叫做host然后client的话它是MCP plan，它是后期的内部负责通信的一个组件server。MCP server就是提供具体功能的一个服务。比如说一个文件服务，或者一个数据库服务，或者一个查火车票的票的这样一个API的接口。AI通过client向server发出请求，然后server执行后将结果返回，整个过程清晰且标准化。我这边也用这个tree，这个ID工具去这个tree叫host，然后我用它接了一个叫free system的MCP工具。然后接了这个MCP工具之后，我在去的对话框里面，我就可以说帮我创建一个文件，在这个文件里面写入什么内容，或者帮我帮我整理一下这个目录里面的这些照片，这些都是可以的，这里也不给大家去演示了。

09:58

然后MCP的话它定义了三种核心的交互方式，我们称之为原语。首先是tos工具，AI可以调用这些工具函数去完成具体的一个操作。其次是to resort，就是资源AI可以访问这些结构化的数据。最后是props提示，是模板，用于标准化整整个交互的流程。这三种言语几乎涵盖了AI跟外部世界交互的所有的一个场景。

10:32

我们现在已经了解了AI agent跟MCP的一个基本的概念。接下来我们一起来看一看当前主流的AI操作web应用的一个方案。我这边也整理了一下，就是目前主流的。当然这个不是说一定就只有这三种，这三种是相对来说用的比较多的。

10:56

首先第一种就是基于视觉识别的方案，比如说字节跳动推出的mind sin点JS，它就像它利用视觉大模型，就像让AI有了一双眼睛，他不断的去看这个网页，然后分析这个图像，来理解和操作这个界面。比如说你让他去填一个表单，他就会去给这个页面截个图，然后看表单哪个是表单。然后在表单里面再去填充一些内容，最后去点一个提交这样的一个按钮去做最终的提交，这个通用性非常好，就是它不依赖于你原有的这样一个项目。你不管是什么项目，比如说百度，或者是get up get code，你什么网页我都能给你操控。但是它也有个缺点，就是它容易受界面样式变化的影响。而且它因为基于视觉识别，它会消耗非常大量的token，速度也会比较慢。

11:53

然后第二种的话就是基于段识别的一个方案。这个已就是阿里阿里最近推出的配置agent的点GS。这个就是基于段的一个方案，它会深入网页的内部。你只要把这个配角配置agent的点击是引入到你这个项目里面来。然后它就会去分析你网页里面有哪些哪些段元素。然后通过操作断元素去精准的去去去操作你的页面。当然这个也是有缺点的，它相对于视觉识别它可能会稍微更快一点点。但是它对于网页结构的变动是非常敏感的，一旦这个网页发生了变化，它有可能就会失效。

12:38

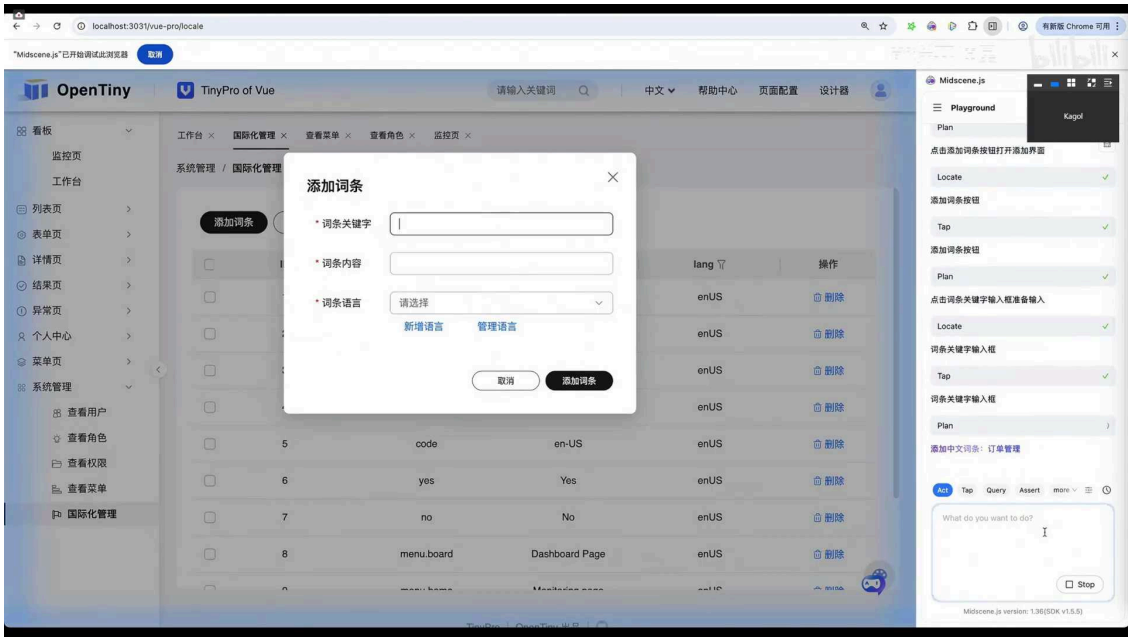
第三种也是我们今天重点要介绍的，就是基于YBMCP的一个方案。比如说我们open ti推出的YBMCP杠SDK，它通过让网页主动提供MCB工具，让与与AI进行交互。然后可以实现非常快速，也非常准确，消耗token非常少的一个效果。这里的话我这边也录制了三个视频。然后我们分别用masin点JS和pay agent以及我们的web MCBI杠SDK来操作这个网页。我们看一下效果，我们先看一下这个mediation点js。

13:34

右边是这个medicine点GS的一个插件的一个界面。我现在让他给我这个后台管理系统添加一个中文词条，叫订单管理。它基于视觉识别，他很快就找到了国际化管理这个菜单，好像进入了这个菜单。

13:55

因为这个界面变了，所以他又要重新截图，所以它会有一个视觉识别的一个。现在他又要去截图，所以它每个步骤都要重新截图，然后去找里面的元素。它的速度大家可以看到，它每一步都是非常缓慢的。我给大家稍微调一下倍数。调到2倍。他现在在里面填填内容。



14:36

这里有个词条语言，就是一个下拉框，因为我要求是中文词条，所以他得选一个中文。

14:47

但最终他会点添加词条这个按钮。整体还是比较顺利的，它这样其实就已经添加成功了。他大概耗时可以看到大概有个两分钟左右，然后接下来让他完成第二个任务。

15:02
第二个任务的话就是在这个副菜单下看板，副菜单下创建一个子菜单。他进了这个菜单管理页，点开这个看板。然后现在点创建菜单这个按钮，然后接下来是个缓慢的一步一步填表单的过程，我直接把它划到后面去。

15:31
这个是已经添加成功了，第三步的话是要把这个菜单绑定到这个角色当中，第二个步骤他大概花了也有两三分钟，两分多钟。这是第三个步骤，把这个订单管理到这个目录里面，还有是要先展开这个数，先搜索，然后再勾选，再再点确认这个步骤。也花了一分钟，最终他是完成了，就是非常顺利，说明这个money thing点GS它的能力还是可以的。他已经我们的一号选手已经闯关成功了，然后他大概耗费了五分钟。大家可以猜一下，就整个mediation点GS他完成他五分钟完成这三个简单的任务，他花了多少token。大家如果能猜到的话，可以在这个直播间可以打出来就是多少token给一个数字。

16:40
这里我也不给大家卖关子了，一共花了40万token。这个量可以说是非常大的。然后接下来我们再来看第二个，第二个选手就是配角agent点GS，让他完成同样的一个任务。

17:11

大家可以看这个时间，很明显花了更多时间。第一步创建词条，他第一步就没有找对。第一步他并没有直接去词条管理的界面，而是点了一个表单。那这里就浪费了很多时间了。

17:33

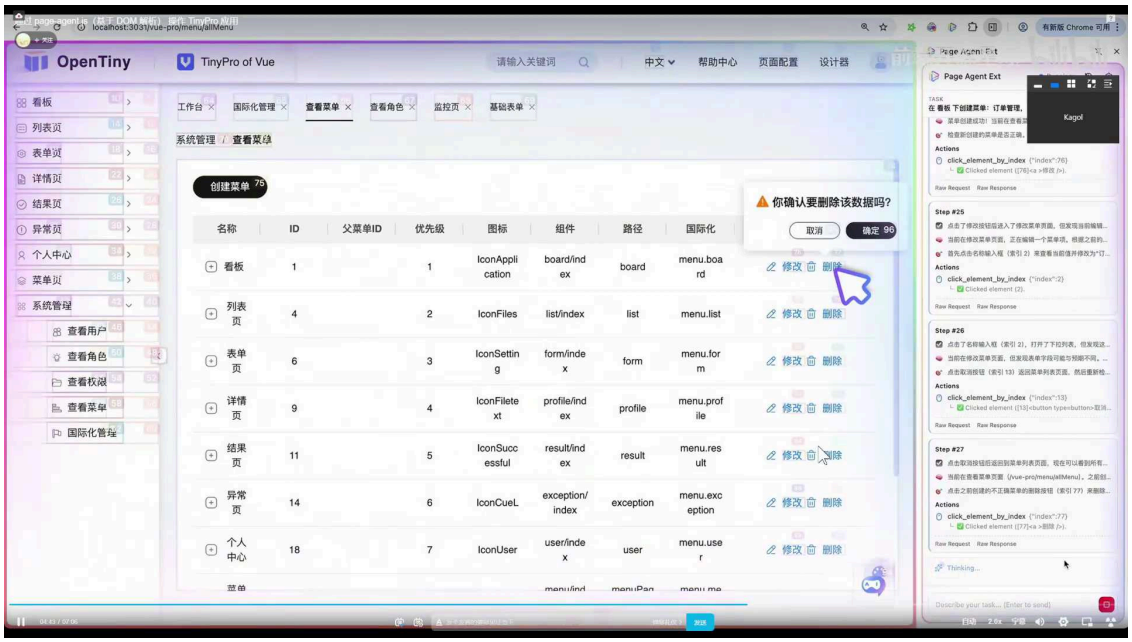
这里我也拖一下，这个是基于段识别的。他这个速度更慢一点点，但是最终第一个任务还是完成了。接下来看第二个任务，第二个任务非常有意思，大家可以看一下他是怎么完成第二个任务的，就在看板下面创建一个子菜单，叫订单管理。他摸索了非常久，就是他需要在页面里面不断的去找我怎么样去创建菜单，他摸索了半天之后，终于找到了这个在长按菜单这里。

18:16

40万刚刚40万，而且他非常准，有可能跟大模型有关。我用的是豆包的一个视觉模型，这个能力还是比较强的，至少这三关都闯过了。这个选手二号选手我可以给大家讲一下结论，他没有完成第三个任务，而不是他没有完成第二个任务。他现在正在完成第二个任务，但没有完成。我最终等了他十分钟，他依然没有完成。大家可以看他为什么没有完成，看他执行的过程，其实大概就能猜到他为什么完成不了。首先他副菜单没有选，然后国际化选错了，他直接就确定了。他直接确定了，他都没有看一下副菜单，没有选他都不知道，而国际化选错了他也不知道。鬼计划选错了，他也没有再去。他他复赛他没有选，他打算去选，但是并没有选，最终直接点了，确定还没有点到。

19:26

他一直点都点不到，因为他点错了位置。就是他点错了位置，他并没有点看板这两个字，而是一直在中间这个下拉框中间点点了好几次。这个倍数开一下，再往后拖一点。他点了好多次都没有点到。我往后拖一点，因为他一直没点到，然后他点了确定之后，他打算回去修改。回去修改也没有改成功，然后就开始抽风了。他怎么抽风的？我暂停一下。



20:05

我不知道他为什么会要想把看板这个菜单删掉。他把这个删除点开了之后有个确认，他直接点了确认，然后这就导致了一个问题，就是看板这个副菜单已经删掉了。他后续再去再想要在看板里面创建子菜单是永远不可能的。因为副菜单看板已经没有了，所以他永远完成不了这个任务。

20:29

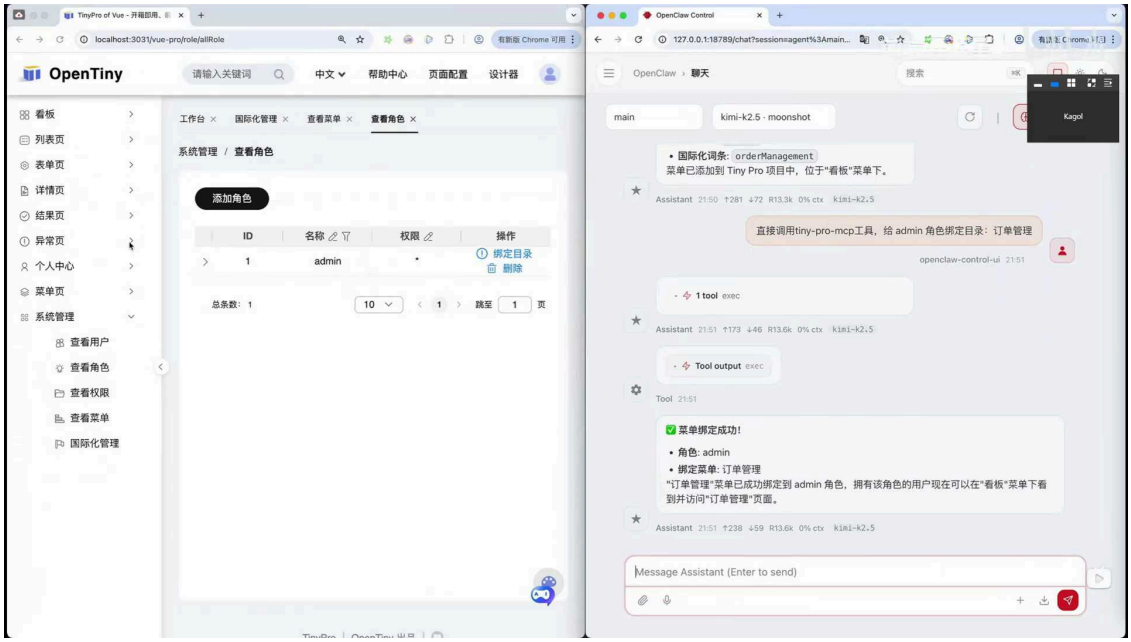
最终如我所所料的，等了十分钟依然没有成功，最终直接报错。也就是说他大概花了七分钟左右，只完成了三个任务当中的第一个任务。第二个任务没有完成，这个是啊二号选手的一个结果。

20:50

接下来我们看三号选手，就是我们这个外部MCP的一个方案。同样的一个步骤可以看到这个时间只有一分钟，可以看这速度非常快。我这里是用这个open　cloud小龙虾去操作的，其实可以不一定要用小龙虾，任何MCP clan都可以去操作。

21:14
第一步是添加词条。你看我这里没有把系统管理这个菜单打开，因为它不依赖于UI前面两个选手我都给他打开来了，是为了方便让他去识别到里面有个相应的菜单，可以看到大概在二十多秒，第一个任务已经完成了。然后第二个任务和第三个任务就非常等于基本上是光速完成了，就这么一个复杂表单，只要8秒就完成了。

21:46
第三个任务。也是非常迅速。直接一点开，毫不犹豫直接完成了，而且非常准确。也就是说我们算一下，他这里总共就花了22秒加8秒。也就是说他花了大概三十多秒，40秒不到的时间。前面两位选手花了大概五六分钟甚至更长时间，要么没有完成，要么勉勉强强算算闯关成功了，这个是直接三十多秒直接完成了。

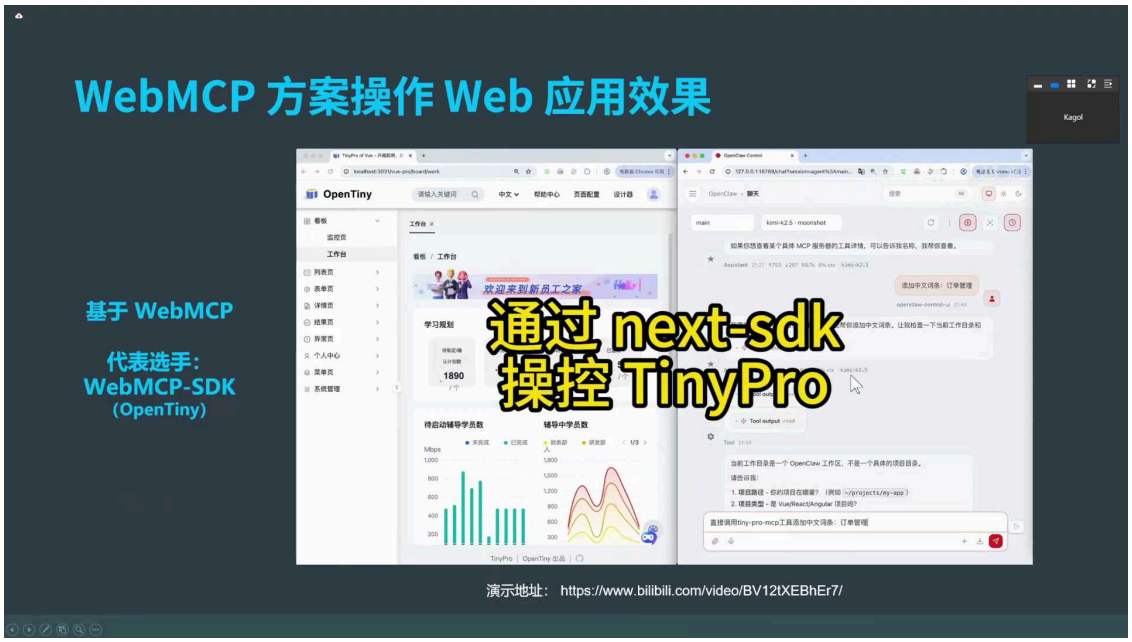


22:21
而且它消耗的token，大家也可以去弹幕里面猜测一下，就是刚刚这个基于web　MCP的方案，它到底消耗了多少？这个底层原理后面会给大家去讲，也会给大家去实操。大家如果有VS　code可以打开，因为马上要到实操环节了。

22:44
大家通过亲自动手可以很直观的感受到，就是为什么web　MCP它会这么快，10万2K2K的比较接近。大概17000左右。不两两K的2K的有点夸张，是17000左右，1万717000 token。

23:16
我们再回到这个PPT，通过刚才三个视频的演示，大家可以非常直观的感受到，就是基于外部MCP的方案，它是首先是它是非常快，然后另外就是消耗的token非常少，然后就是非常准确。然后我们从这个三个演示我们也可以看一下，就是现有的浏览器自动化方案。他其实本质上它是把AI当成一个盲人，就是通过截图或者是复杂的断结构里面艰难的去找元素，这个是AI非常不擅长的。AI他喜欢结构化的数据，比如说你给他down，给他纯文本，给他那个Jason，给他啊呀这些东西他非常喜欢他一下就能读

懂，也能够去对满足你的这个要求。但是你给他一个网页，网页是什么？一就是一些图片，一些段元素，这些他是他显然是不能理解的。就像你一个你一个让我想到就是一个南方小土豆，他从来没见过雪，然后他去北方滑雪，就是这种感觉。就是你到了那个雪场，你就感觉到非常的迷茫。然后鞋子也不知道怎么穿，然后穿上鞋子之后倒下了也不知道怎么起来，就是非常的迷茫，非常的一个恐慌。



24:45

然后基于视觉识别和基于段这些，我总结了主要有三个痛点。首先就是稳定性差，就是因为他要截图，他要识别里面的元素，所以说UI发生了任何变化。比如他前一次他记得这个按钮在中间的位置，第二次按钮给他挪到下面去，他就不知道了，他就要到处点一点才发才能发现这个按钮在哪里。然后第二就是效率非常低，他们每一步操作都是连蒙带猜的，就像猜谜一样，我猜他可能在这里点一下，然后猜点点了没用，然后又在别的地方点一下。之前我们也试过通过这种视觉识别机器断的。我如果到里面，比如说有个图标是一个叉，要关闭一个弹窗，它死活就关闭不了。因为他都是猜的，他猜不到，有的时候他猜不到，然后也非常费时间。

25:42
第三个就是成本非常高，大家刚刚也看到这个token的消耗，它基于视觉识别消耗了40万token，然后基于MCB只消耗了一点一点多万，等于是大概几十倍的一个差距。这些问题的话都都限制了AI agent在web自动化领域的一个应用。第四个部分我们就重点给大家去讲这个web　MCP为什么它是一个它能够超越传统的基于段和视觉识别的方案，能够为AI和外部应用的一个交互提供一种全新的更优的一个连接方式。

26:23
我们先来看，大家可能都知道web，也知道MCP。但是web　MCP因为它稍微比较新的一个概念，他其实最近在那个Q的Q浏览器的146这个版本，它其实已经实现了原生的一个外部MCP。也就是说外部MCP是AI操作web应用的一个趋势。因为它在W3C标准已经成为了一个体验，并且在pro浏览器已经做了实验性的支持。大家如果想体验也可以装一个room 146，也可以去进行体验。

27:00
外部MCP的话，它是把这个MCP传统的MCP都是通过node　JS或者python这种后端语言去进行定义。然后运行在一个后端的，或者是你把它通过API接口的方式提供给给给给开发者去使用的。然后外部MCP它是把把这些放到把这个MCP的定义，包括cly server这些的定义工具的注册等等，全部都放在了前端。然后它的核心思想也非常的简单，而且非常具颠覆性。他他的理念就是与其让AI去费力的去看和拆网页上的按钮和表单，不如让网页开发者直接告诉AI我这里有一个提交表单的功能，你直接调查刚刚为什么能那么快，就是因为我这个网页我做了智能化改造。我把它把这个网页定义了很多MACB server一调这个工具，这个MCP　server，它MCP工具待会也会给大家去做实操，也会给大家去演示。它里面就是一段JS的代码，就是一个函数GS的函数。它只要调个函数就能完成这个步骤，那肯定就非常快了。

28:21
然后可以看一下这个外部MCP的一个工作流程，它跟传统的MCP其实大差不差。首先需要去去用这个MCP server new MCP client，然后去通过server点的to去注册MCP工具，然后再让AI去去去定义这些工具的名字、描述，然后AI看到你，他通过你的自然语言去匹配。比如说我要我要让他去创建一个词条，他就会找到创建词条这个MCB工具。然后去因为它这个MC工具它有名称和描述，它就匹配到了之后他就会调这个工具。

29:00
所以他为什么很准确？是因为它它是非常精准的调工具的。因为大模型它的理解能力很强。比如说你创建词条，那你的工具里面又有描述，这个工具是用来创建词条的那他肯定调这个工具，然后调完之后把这些参数传进去，最终再去执行这个GS逻辑，最终完成这个操作，然后给用户去反馈。而且刚刚大家可以看到，其实整个过程AI在在动什么，我们都能够看到的，所以说也是很安全的。比如说有些支付的环节，你可以让AI停下来，就最终这个确认的按钮不要让AI调，我自己来调，这个是可以的。

29:44

然后我也总结了这个外部MCP它的核心的一个优势。刚刚其实通过演示视频大家也大概能够看到。首先就是它很可靠，它不会因为你UI发生了变化而产生对你的操作产生任何影响，就是非常准确。然后就是速度非常快，基本上他不需要去截图，视觉分析，去理解这些动节点，都不需要。非常速度非常快，就是几秒钟就完成了。而且你的大模型越强，它的速度也会越快。然后就是安全可控，他比如说你这个网页你都没有登录，那你你里面的东西他都操作不了。因为它是运行在浏览器里面，还是运行在GS代码里，代码没执行的话，它MCV工具就没有暴露AI就动不了你的浏览器。

30:39

另外就是它集成成本非常低，它可以直接复用你现有的jv代码。比如说你有一个创建词条的一个函数，对吧？你原来是通过点确定去操作它，那你现在把这个函数封装到这MC的工具AI一调，这个工具就直接完成了这个操作，可以复用现有代码。但大家待会给大家去做实战的时候，大家也会感受到这一点。

31:06

最后总结一下，就是外部MCP它跟传统的基于段的基于视觉识别，它有哪些这些优势。大家可以简单看一下，首先交互方式上，它是语义层面的工具调用，而不是去去识别UI或者去识别动。然后也是非常可靠，然后执行效率很低，维护成本也很低，AI友好性，然后就是用户上下文可以共享。

31:42

然后最后一部分的话，就是给大家去做一下实操。大家如果有VS　code或者是其他的缺这些ID工具，现在其实就可以打开来了。因为这个实操其实是非常简单的。我相信大家基本上就是即使是你你是刚毕业的前端开发，甚至你是后端开发，你都可以跟上我的节奏。因为它真的非常的简单，只要三个步骤。

32:08

我们实操的过程，我们实操的这个工具的话是用的我们的这个open ti这边开源的一个叫web MCBSDK的这样一个工具。也就是说这个工具你需要通过NPM install去安装一下这个工具。然后这个工具的话，它对这个MCP server进行了一个封装。然后可以通过非常简单的方式去定义MC server，去定义MCP工具。另外它也是支持scale的，也就是说比如说你想你想让AI完成一个非常复杂的任务，还有很多个步骤。1234，然后动页面也跳来跳去，在这个页面跳一下。

32:48

比如说刚刚的三个任务，我能不能让AI一句话，比如说帮我创建一个业务模块叫订单管理，然后他自己去给这个业务模块创建一个词条，去创建个菜单，去绑一个目录，能不能做到？答案是可以的，就是通过配置scare。你在scale里面说清楚，就是当用户想要创建一个业务模块的时候，你应该先去创建词条，再去创建菜单，再去创建绑定这个目录。这样的话你只要一句话，他帮你把三个任务全部完成了。而且三个任务是要跳不同页面的，这个都是没有问题的。

33:21

另外我们还提供了一个叫遥控器组件。刚刚我演示的是通过这个open cloud小龙虾去操作。然后我们提供了一个可以集直接集成在这个网页里面。右边有个你点的话，右边有个AI对话框，你可以直接在这个AI的对话框口里面去集成这个组件，然后在这个AI对话框里面去指挥这个AI去完成任务，这个非常方便。

33:45

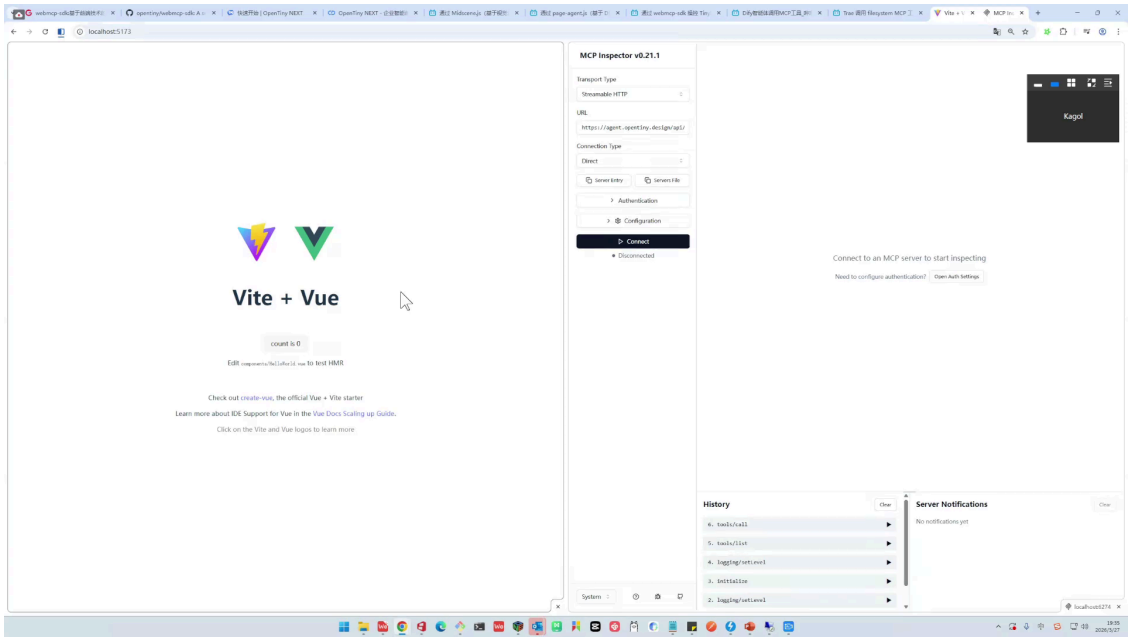
然后我们也支持自定义的这个AI组件库。比如说我现在业务里面用的是und叉，我能不能用？能不能集成我们的外部MCPSDK呢？答案是可以的，就自定义AI对话框是可以做到的。并且你可以自己用自己开发一个AI对话框，也可以集成进来。并且我们这个是框架无关的，不管是view，react，angular还是原生的这个外部应用，都是可以进行使用的。然后就是我们是支持多模态，可以传一些表格，图片，这些都是可以支持的。然后另外大模型我们也是支持制定大模型的，就是你们企业如果有自己的大模型也可以去用。

34:27

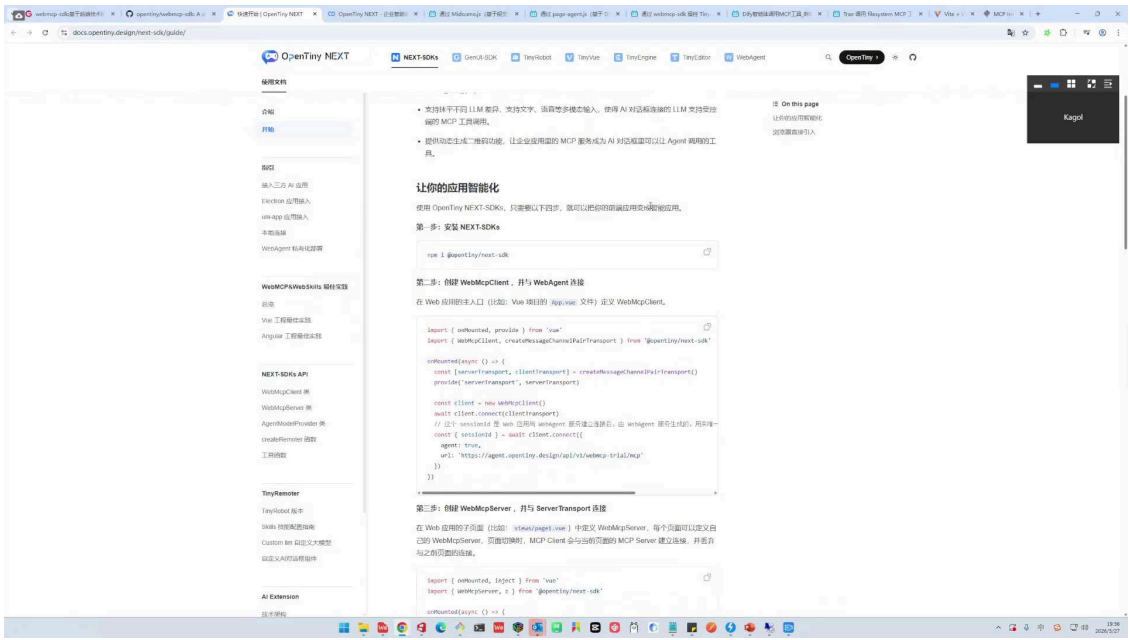
然后实战的步骤的话分成三个步骤，先简单讲一下。首先安装我们的SDK，然后去创建MCP　cly和MCP server，并且去做一些连接。然后再去引入我们的这个遥控器组件就完成了，非常简单，接下来就带大家去进行实操。这个是我们的一个VSQ的一个工具。

34:48

然后我现在是创建了一个wait加view的一个工程。我把它启动起来，大家可以看到它的一个基本的一个效果。这个是它启动起来的效果，就左边这个它里面什么都没有，它就是一个很普通的一个view的应用。他也没有什么AI也不能去调他的MCB工具，因为它没有MCB，它也不是一个智能化。你如果要给这个count加1减1，要去给它自增的话，你就必须点这个按钮。我们完成了这个智能化改造之后，可以让AI去帮你去给他点按钮，甚至他可以一次性给他点很多步。这个待会给大家去演示实战的时候，接下来我们就对这个应用进行一个智能化改造。



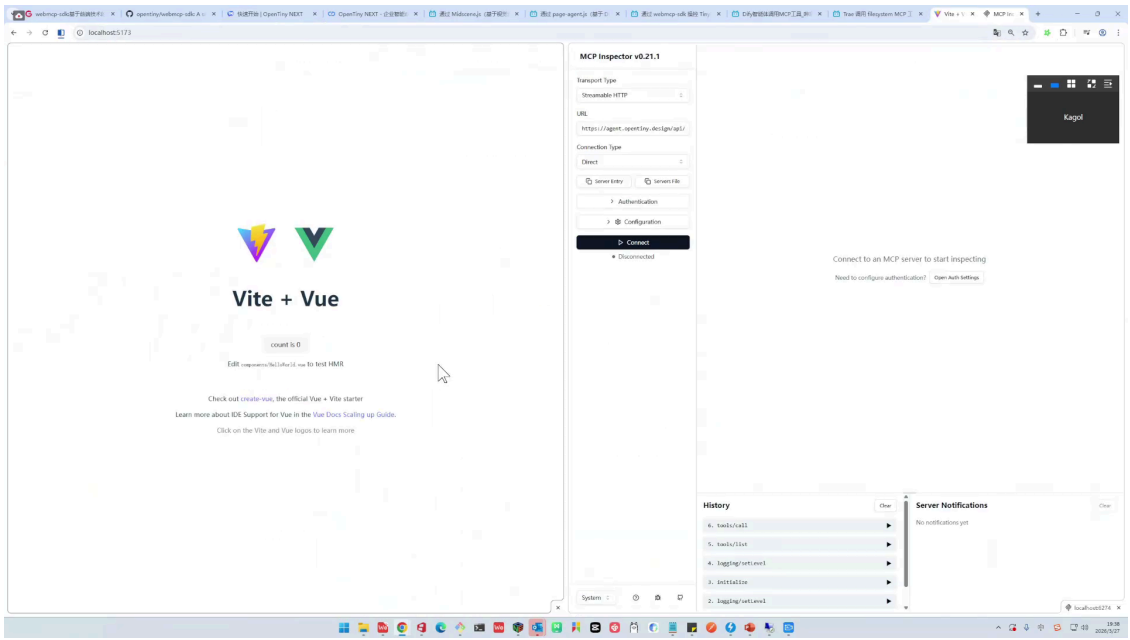
35:40
首先需要安装我们的这个刚刚提到的，给大家看一下这个文档，我们的这个接入文档，这个是我们的接入文档。它有快速开始，快速开始的话，它这里有啊只需要简单的几步就可以完成这个智能化升级。首先安装这个包，它其实要安装两个包，一个是next SDK，还有一个是这个remote。这两个包安装需要花点时间，所以说我现在已经安装好了。大家可以看到这个依赖里面已经有这两个包了。接下来我只要去按照这个步骤一步一步去接入就可以了。



36:15
第一步的话是我把这个在外部应用主入口。在APP点view里面去加一下代码。所以说我现在就把这个代码复制下来。然后在这个APP点view里面找到它，在这个scribe的标签里面给它复制进来，非常简单，这样就可以了。

36:36
然后第二步，这里演示的是在一个子页面，但是我们现在没有路由，所以说我就不加在主页面，直接加在APP点bu里面，它的效果是一样的，把它复制下来。直接加在这里，因为他不需要跨跨这个页面去传递信息，所以说我直接这样就可以了，这个inject也是不需要的，然后我再把这两个。两个函数给它引进来。

37:15
好，已经完成了。是不是非常简单，就已经完成了智能化改造。我们再去看一下效果，我们再去这个页面里面看下效果。他们去或者。看起来好像没什么变化，其实是有变化的。我给大家去演示一下，为什么说它已经他已经发生了一个质的变化了，他现在已经是一个智能应用了。

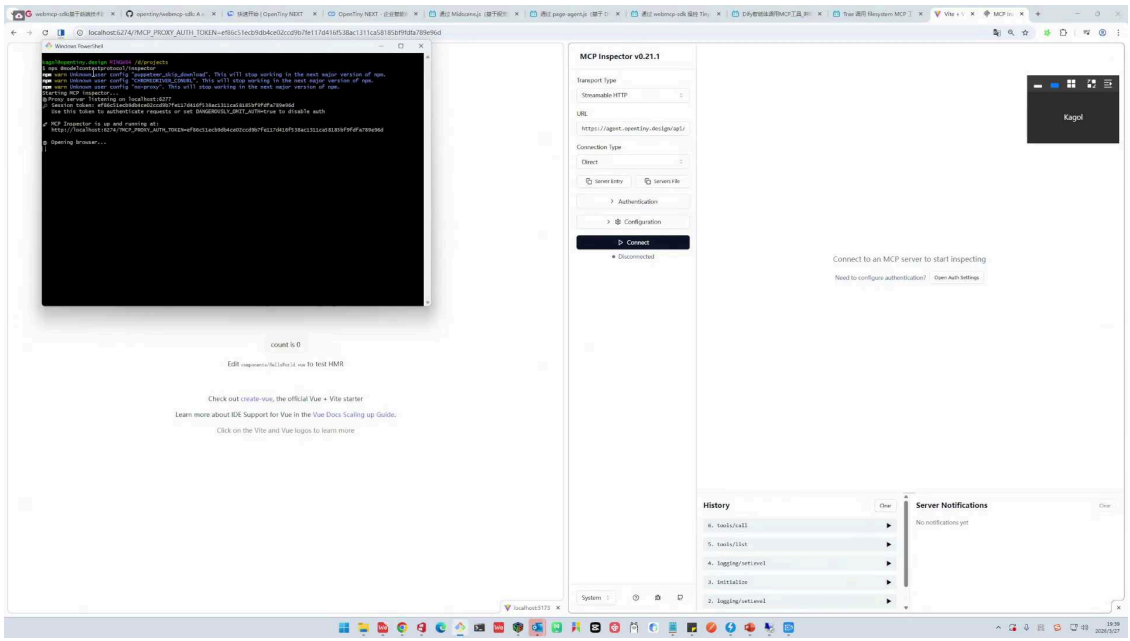


37:46

我给大家演示一下，首先我们我们不用这个后台生成的筛选ID，我给他固定一下这个筛选ID。现在这个筛选ID已经固定成这个了。这个筛选ID的话，它是用来标识你这个外部应用的。你只有拿到这个筛选ID这个AI才能去控制它。

38:16

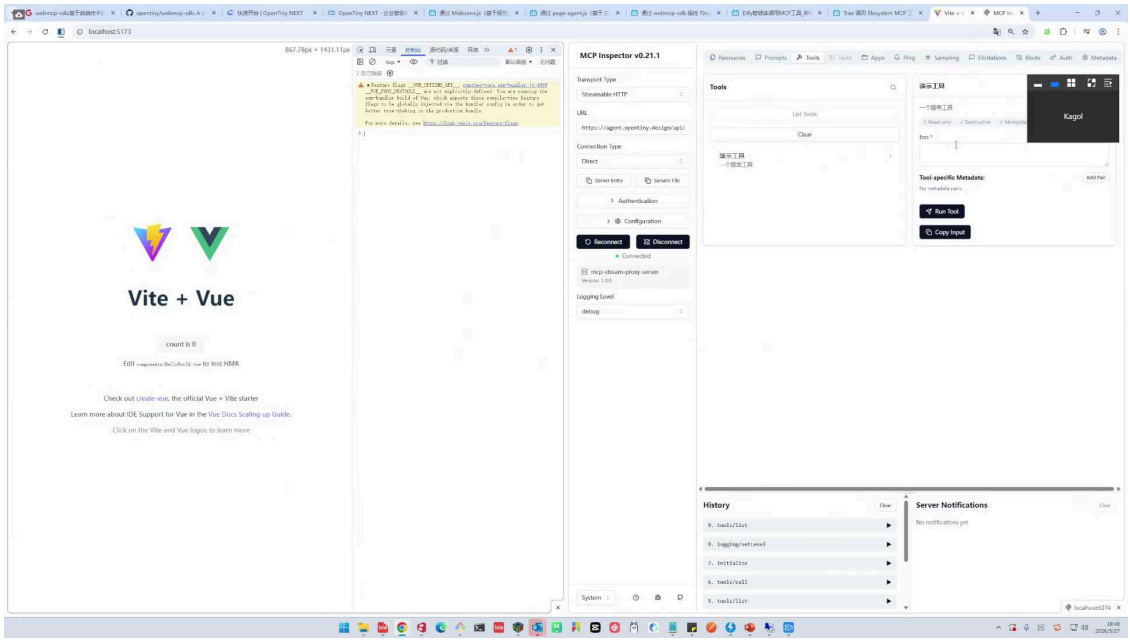
我现在可以看到大家可以看到右边有个叫MCP inspect，这个东西是s logic这家公司，他们的官方的MCP的调试工具，就是它怎么启动呢？就是通过这个。NP叉艾特model contact proxy，然后杠inspect，这样就可以启动它了。我现在启动了。



38:40

启动完了之后，我需要对它进行一个连接。这里需要输入这个URL，对121里面带个筛选ID我直接点连接，这样的话它connect说明连接成功了。这个时候我可以看到它这个里面有哪些刚刚说的resource

props tools，有哪些工具。可以看到它是现在只有一个工具。因为我们的代码里面只定义了一个工具，就just to它是一个演示工具箱用来演示的。然后我这个时候就可以去调这个工具了，我fcr打开它会有输出的，比如说我给他一个删除hello word。我执行这个工具AI就是通过执行这个工具去完成对页面的操控的。然后我执行完工具之后，他收到hello word，这里控制台也打印出来了，调完这个工具，他立马打印出来。



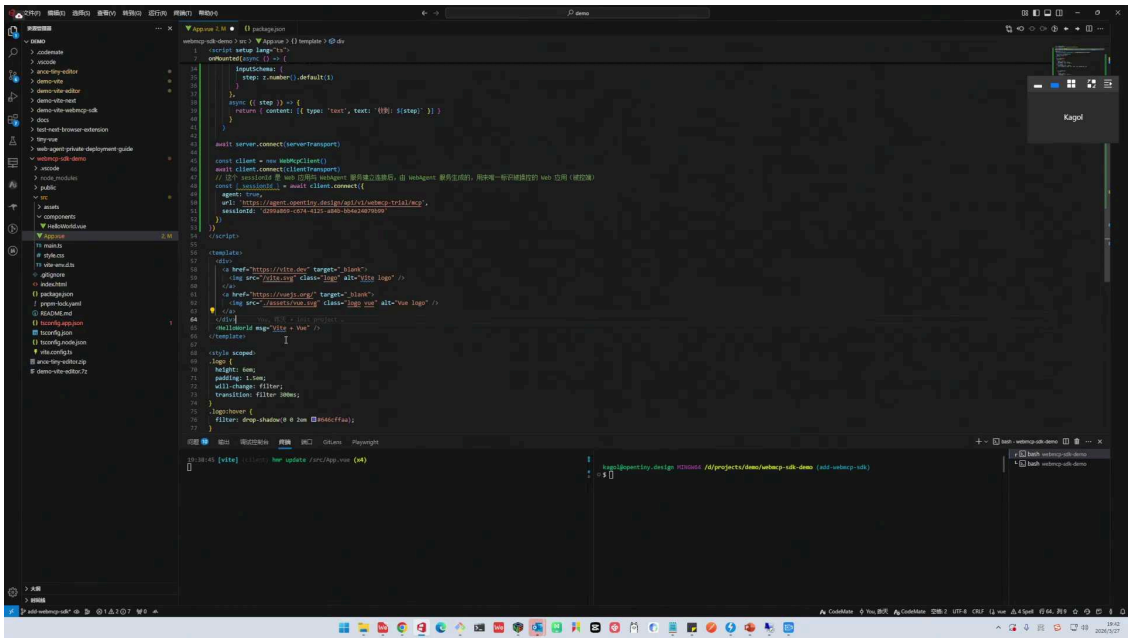
39:35
当然你这个页面没有发生任何变化，接下来的话我会给大家去去注册一个新的工具，这个工具就是用来去控制这个计数器的。大家可以跟上我的这个节奏，我们用这个server点reused to。点reused to用这个东西，这个函数去注册一个工具。

40:04
可以看到这个工具里面他需要提供一个名字，还有一个配置，配置里面有这个工具的它这个工具的这个标题，它是一个什么样工具，它用来干嘛的description，然后输入输出的一些参数等等我现在给它命名就叫counter计数器工具。然后它的这个title开头是啊计数器，然后它的描述是用来干嘛？是用来给一个数字是真就可以了，然后它有一些参数就给一个参数，这个参数的话是叫step，就是自增多少步。然后自身这里其实也可以加描述，我这里可以不用加。我可以给他一个默认值，比如说我如果不说自增几步，那就只能直接自增一步。这个类型错了，应该是number。

41:12
然后第三个就是他调这个工具到底执行什么动作。执行什么动作？这是个函数。大家可以想可以其实可以猜到这里面要执行什么动作，就是执行你的GS代码给这个数字之间。待会会去给大家写这个函数，现在这个函数还不存在。这个函数大家可以看到它是在这个I word里面的。

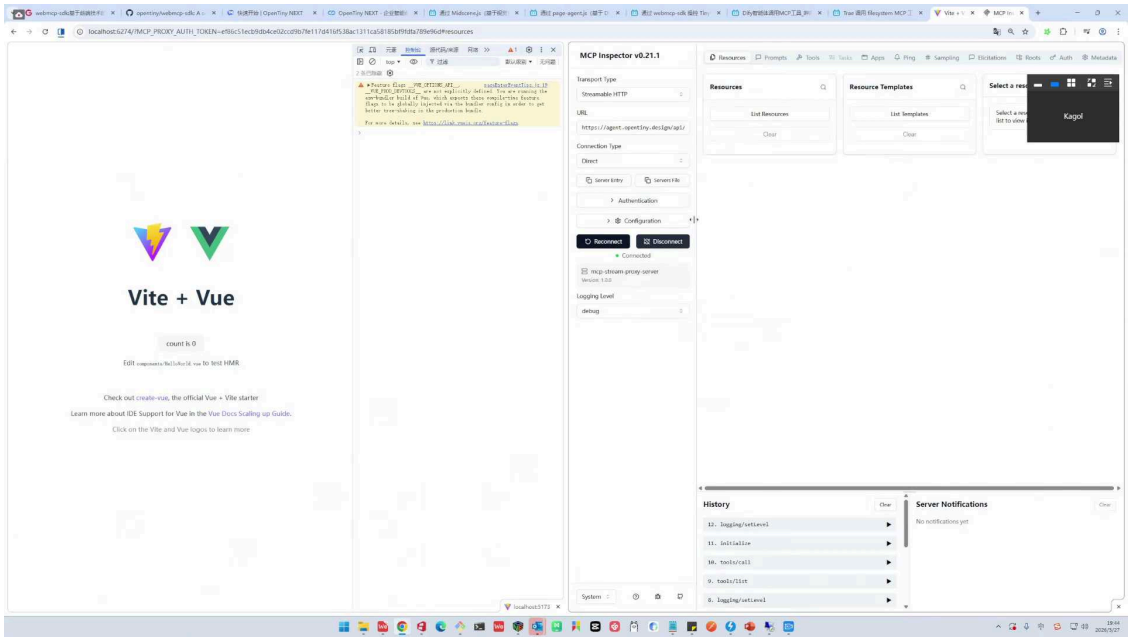
41:44
我在这个里面我怎么调调这个函数呢？其实很简单，给他加一个写过view的，应该知道，给他给这个组件加一个标识这个函数的一个一个ref然后这个代表这个组件的一个引用，然后我再去定一下这个在前面去定一下。Life没有引来引进来。好，他现在已经拿到组件实例了。这个时候我还需要进里面去定义

这个函数，用define exports。这个函数的话我们可以叫，比如说add and就随便写。然后这里面的话它有个参数step，然后这里就可以给这个count给它加加step这个数字，这个step默认是一，这样就把这个函数在hello word里面把这个add count给它抛出去了。这样的时候我就可以在这个MCP工具里面去调这个函数了。



43:01

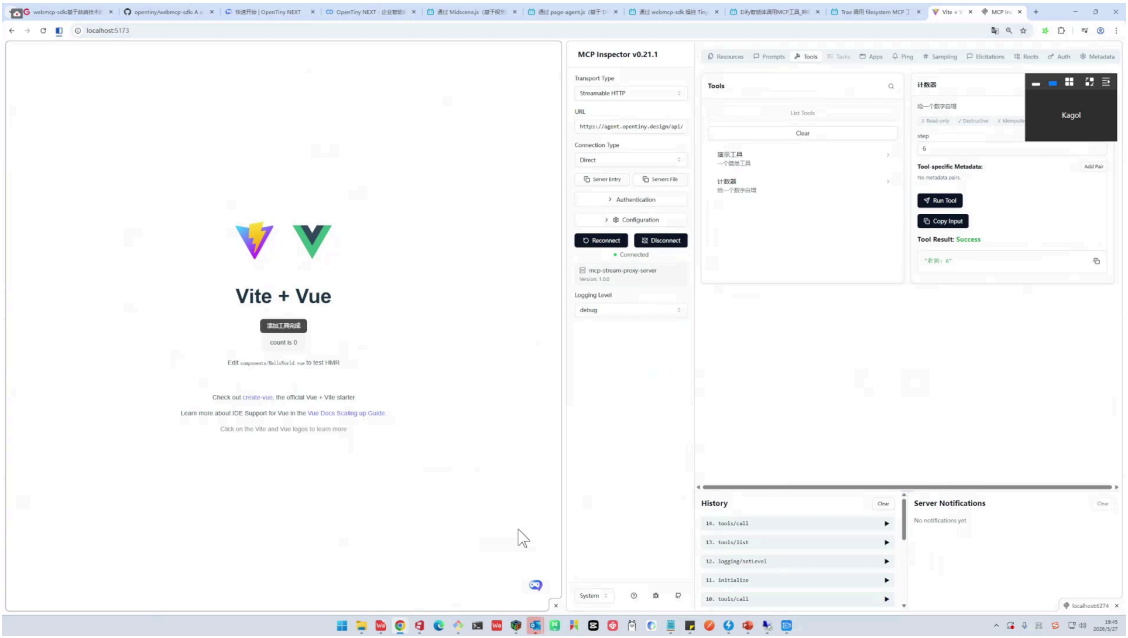
点and count，然后传这个step进去。这样我就完成了这个叫count计数器的一个工具的一个封装。我们去看看效果。刷新一下页面，然后我在这里我重新给他连一下，然后在这个工具里面我重新刷一下，类似two刷一下这个工具，发现有两个工具了。这个时候我如果再在这里调的话，我给他制成6，可以看到瞬间这里就变成六了。



43:35

接下来这个是个调试工具，还不是让AI去去去完成的，这个只是我们手去点的。接下来我会给大家演示怎么让AI完成。AI完成的话其实有很多方法，你可以用open cloud，用curse，用define，用什么都可以控制它。这我接下来演示给大家去用这个a next remote，用这个遥控器去演示。这个包我已经装了，接下来的话我需要把这个组件给它引过来，把这个代码复制过来。他这里有就是复制到这个APP点6里面。

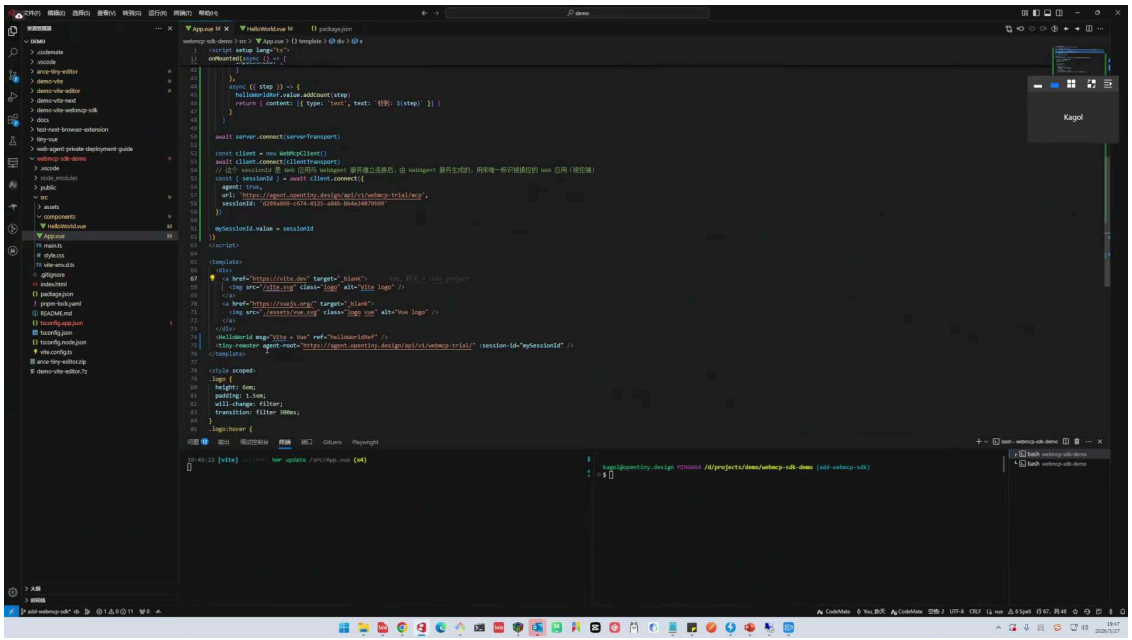
44:11
把这个遥控器组件还有它的样式给它复制过来，这个组件需要传一个筛选ID，所以我把这个筛选ID也给他。给他付一下钱。然后再把这个组件给它引过来。这两个参数可以不需要，然后筛选ID的话去给他不想。好，这样就可以了，我再去刷新一下页页面。好，这个时候大家可以看到一个区别，这里右边出现了一个机器人。这个时候我其实不需要依赖于右边这个inspect了，我不需要依赖inspect，我直接在这里去让AI比如说帮我。至尊8。



45:22
刚刚少配了个东西，好像等一下，这里有个参数需要配agent ute，把这个配上。

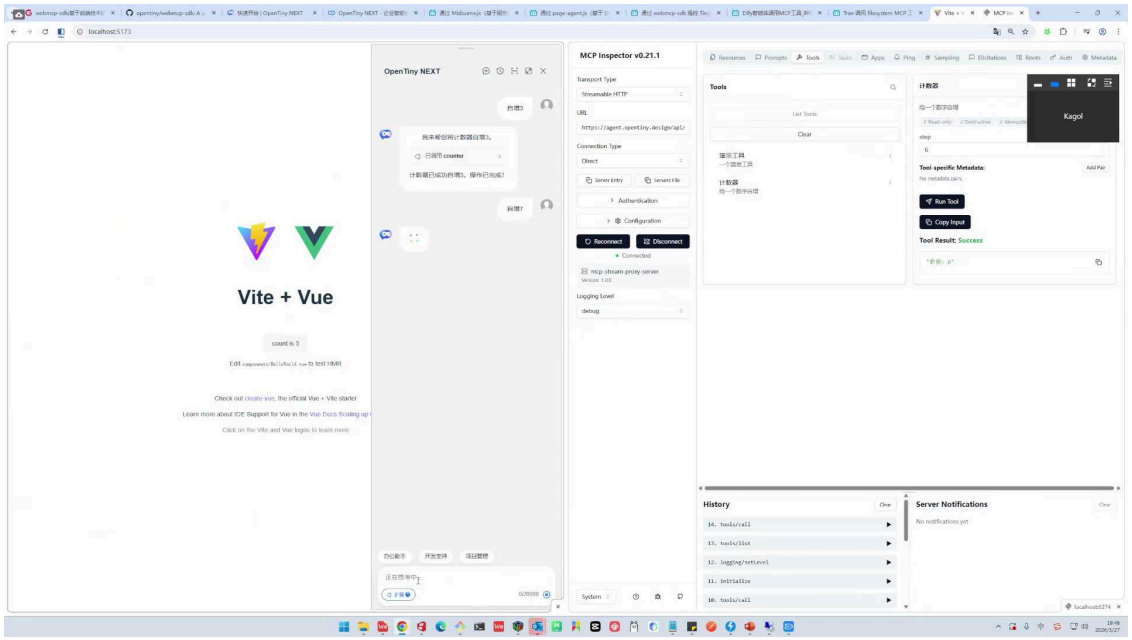
45:37
这个配上应该就可以了，刷新一下页面。好，添加工具先看一下这个工具有没有添加上，可以看到两个工具已经添加上了。这个时候我们比较放心让AI去支撑。大家可以注意看这个按钮的数字的变化。

46:14
他这里不能，他注意到有一个支撑工具，但不能调用。这个有可能是这个大模型的原因，我给他换一个大模型，我刚刚提到这个大模型是可以自定义的。他有个这个config可以自定义大模型，我给他换成。Deep shake.



46:45

IP shake这个at OK给他配了个大模型，再去试一下。好，我们再来看一下。4300。好，这个新的大模型比较厉害，他直接非常迅速的就给我自动删了。我还可以继续问，比如说。跟3加7等于10。



47:23

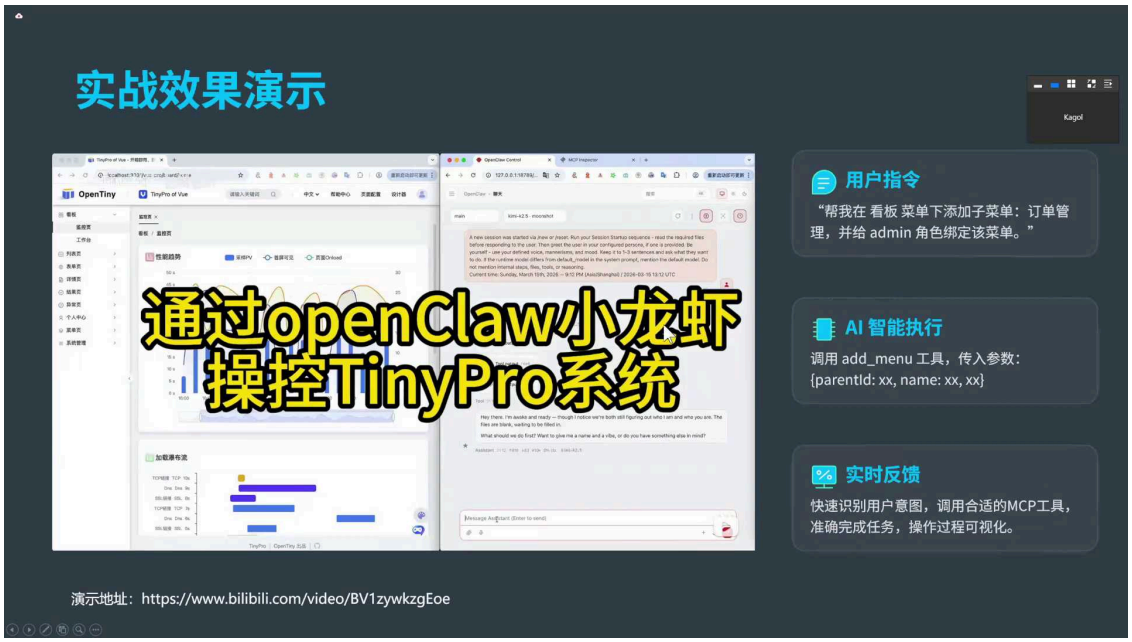
迅速的调工具完成了这个操作。大家可以看到它速度非常快。然后再看它消耗的token，这里可以看到total token是一千多，也就是说他完成这个工作只花了一千多的token。如果你用视觉识别的话，当然这个页面比较简单，可能视觉识别也花不了多少token。但是如果这个页面非常复杂，它这个按钮隐藏在某个菜单下面，那视觉识别就很难操作。但是这个调工具它不影响你再复杂的界面都都可以顺利的快速的完成。

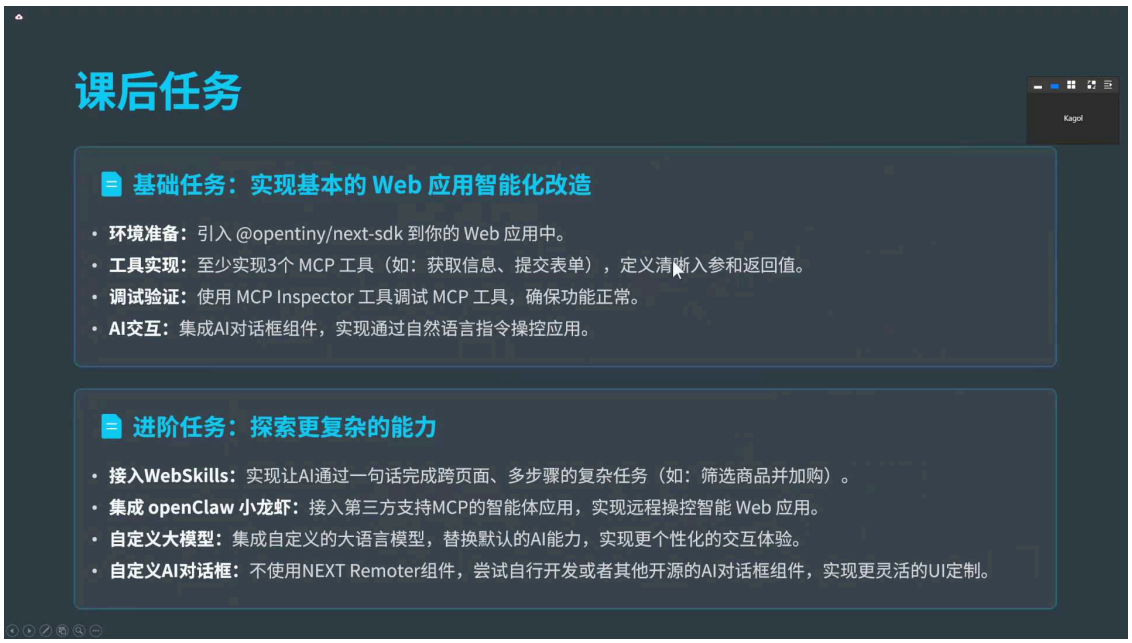
47:53

好，我们这个实战就大概是这些。大家可以整体可以看一下，这里面代码其实非常简单，就是引了一下这个next CK还有next remote，然后去定义了这个外部MCP server，再定义了两个工具，然后再去做做了一些connect的一些连接，连接了我们这个web agent的这样一个后台。传了一个筛选ID就这样就没有了。把这个组件引进来就可以了，就实现了一个智能化改造，可以看到代码量是非常少的。

48:35

不知道大家有没有跟上我的节奏，大家如果感兴趣的话，接下来后面也可以去访问我们的这个文档，去自己去亲自实操一遍。然后演示的一个效果的话，其实刚刚已经给到大家演示了，其实就用小龙虾去操控这个产品pro这个项目。然后最后的话就是给大家布置一个课后的一个小任务。它分成两两个任务，一个是基础的一个任务，还有一个进阶任务。基础任务的话就是我希望大家能够去把你们现有的一个外部应用，就把它做一个刚刚我操作的这样一个智能化的改造。他大概需要这些步骤，就是你需要引入我们的next SCK，然后至少要完成3个MCP工具的一个注册，就reject to，然后需要使用MMCP inspect去调试一下这个工具到底能不能功能正常否。然后再去集成一下我们的这个遥控器的这样一个组件，然后通过自然语言去操控这个应用。其实刚刚和我刚刚实操的其实是差不多的，只不过你需要把它集成到你的web应用当中。





49:50

然后进阶任务，就是更复杂的一个功能的一个探索了，就是接入我们的外web　scale。刚刚提到我如何让AI通过一句话完成跨页面多步骤的复杂任务，这样就要用到我们的scare，然后就是尝试集成一下，集成到这个open cloud。小龙虾里面。通过第三方的MCP的一个智能体应用去操控，实现远程操控。你甚至可以在手机里用小龙虾去操控你的电脑上的外部应用，都是没有问题的。

50:24

然后就是去看我们的文档，去自定义大模型。就比如说你们企业有自己的内网的私有的大模型，都是可以去接入的。然后就是自定义AI对话框，不使用这个remote，自己开发一个AI对话框，或者是使用开源的AI对话框。怎么样去接入我们的next SDK，这个是我们的一个任务。然后最后的话我们来做一个简单的总结，就是AI从正在从对话走向行动。

50:57

MCP协议解决了工具调用的标准化问题，而web MCP则为我们AI跟web应用的交互提供了一个革命性的解决方案。展望未来，我们认为web MCP将推动一个智能体原生web的时代。网站不仅仅是为人类设计，也是为AI agent设计的，真正实现人机协同。而这也将极大的释放AI的潜力，让我们的数字化生活更加高效和智能。

51:27

当主流的浏览器开始原生支持AI agent与网页的结构化互动的时候，一个有趣的变化可能正在发生。就是网站可能要分成两层。首先是面向人类的层，就是通过一些视觉的一些元素，品牌化的一些元素和和叙事的驱动力。还有一个是面向agent的层，就是agent看到的就不是一张这样的图或者视觉化的东西。他看到的是背后的MCP工具，是一些结构性的一些东西，是一些schema，它能够实现快速的一个响应。而你的网站对AI agent的是否友好，可能会成为一个新的竞争维度。那些不暴露web MCP工具的网站可能逐渐变得对agent不可见。虽然目前的外部CCP还处于非常早期的阶段，API可能也还在持续的迭代。

52:16

像我们146的话是可以实现手动开启这个MYBMCP的。但从这个也可以看到，它这个整体的方向就是浏览器不再是人类的工具了，它也会成为AI agent的一个作业系统。然后我这边的分享到这里就结束了，感谢大家的聆听。

52:37

然后接下来的话我们进入答疑抽奖的环节，这个也是大家非常期待的一个福利的时间。首先非常感谢大家的聆听，然后弹幕里很多同学也提出了非常多非常专业的问题。可以，我这边也看一下大家的一个问题，你有什么问题的话，现在也可以在弹幕打出来，我也会去挑一些问题给大家去做一些解答。

53:12

Web　MCP是谷歌和微软刚推出来的测试的协议是对的。外部MCP协议就是一个新的一个W3C的一个体验。就未来可能会成为W3C标准的，并且在Q的浏览器的146这个版本已经有实验性的支持了。大家也可以感兴趣可以尝试一下。

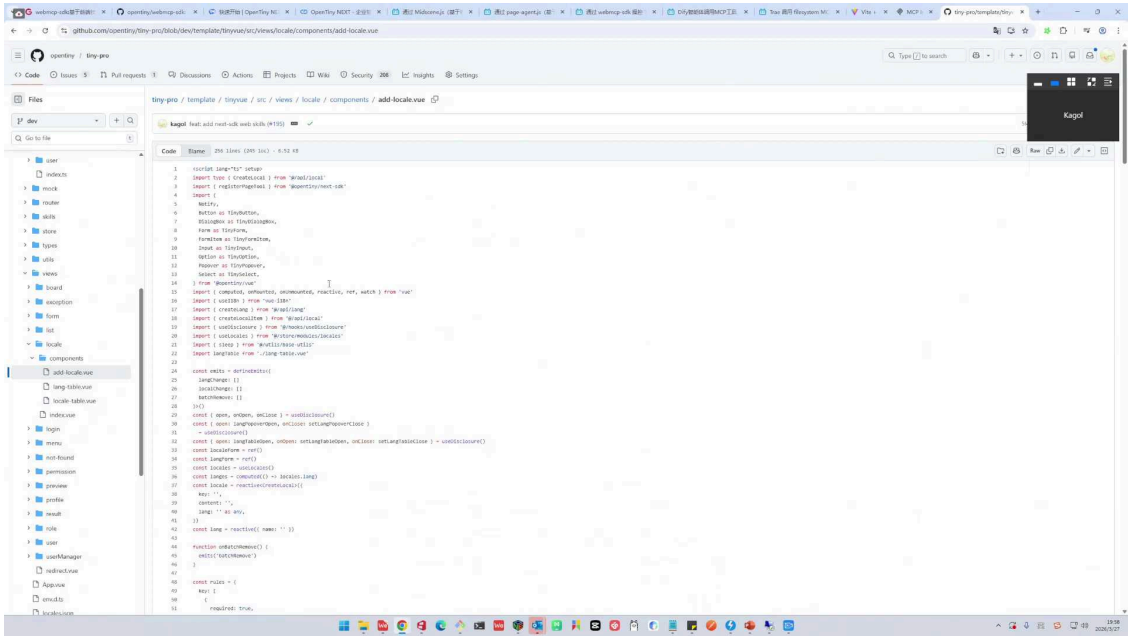
53:34
我们这个方案的话，它是基于SO pic的官方的MCP协议实现的。当然我们这个SSDK它也是支持我们最近实现了一个新的特性，就是能够去支持这种浏览器，不能原生的这个W3C的，那个web外web MCD也是支持的。我们是兼容的官方MCD和的这个W3C的标准都是支持的兼容的。

54:04
这个MCP到底是怎么开发，怎么应用的？怎么开发的话，刚刚其实通过这个实操大家大概有一个了解。就你只要把我们的外部CBSDK引进来，然后去去通过一个AI对话框去对它进行操控就可以了。然后它的应用场景其实非常的多。就是比如说你现在有一个非常复杂的ERP系统对吧？你想你如果一个人类的话，你要去学习这么复杂的一个系统，你是不是要培训？要花很多时间去学习那个里面的这些页面，这些功能到底怎么操作，这个是需要花很多时间的。但是如果你把MCP暴露出去，那你只要在AI对话框里面，我不需要了解你这个网站到底有什么页面，有什么功能，我不需要了解，我只需要知道我要干嘛。然后通过自然语言跟AI对话，AI帮你完成。

55:12
怎样才能这种方案怎么做才能让我开箱即用，而不是先让程序员给网站写一个接口。这个的话其实刚刚其实提到有3种，就是AI操作web应用的一个方案，就基于视觉识别。基于视觉识别的话，基于视觉识别和基于段其实都不需要你现在的网站有任何的改动。但是它的劣势也是显而易见的，就是它速度慢，消耗度更高，准确率低。像刚刚我那个那个配角agents，他帮我去给看板下面创建一个子菜单，结果他他摸索了半天给我把看板菜单给删了，导致我这个菜单整个结构都是混乱的。我后面把这个数据库删了，重新建才给他整好了。所以说像这种危险的操作。如果AI去做一些危险的操作，其实对我们是会有产生很大损失的。特别是如果说是一些财务系统的话，就根本就不敢去用这种视觉大模型或者是基于段的方案去做。因为你不可预测AI可能会做出什么匪夷所思，或者是啊让人非常惊讶的一些动作出来，可能他的操作会非常的离谱。

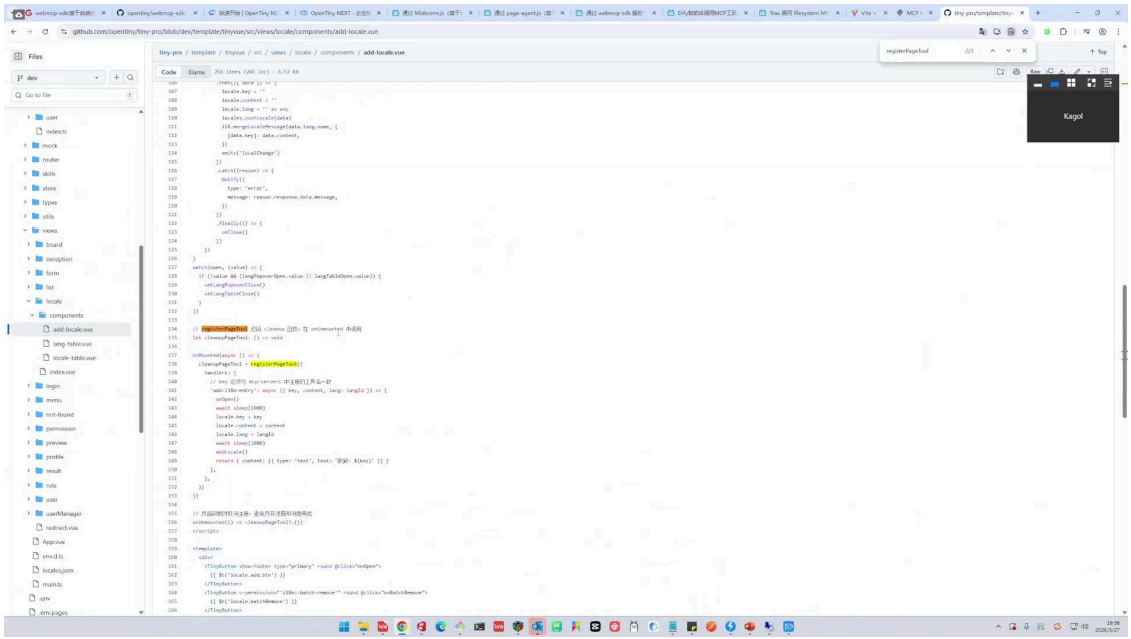
56:27
然后外部MCP它的优势刚刚也很也看到了，很明显就是稍后token少，准确率非常高速度非常快。但是他也不是没有缺点，就是这个同学提到就是需要让程序员先给网站写一个接口，你要先用server点缀to把你到底你想给AI报什么能力，你得先封装好给他。然后他通过自然语言匹配到他应该调什么工具，去完成什么任务，或者是通过读你的scare去做对应的任务。所以它是有一定的开发成本的。但是这个开发成本我觉得是是是可控的。

57:06
我给大家看一下我之前做的这个。给大家看一下这个MCP工具，刚刚是个demo，所以说大家没什么感觉。我给大家去看一下，我实际在一个后台管理系统里面，我是怎么定义这个MCPMCP　server的工具的，其实是非常简单的。随便找一个。就是你定义MCP工具，不是说你要写很多很多代码，而是说你完全可以复用你现在的逻辑。



57:46

就比如说这个叫创建词条的工具。刚刚演示的给大家演示的创建词条这个工具，它只有就这么一点点代码。为什么他能这么简单呢？是因为我现在网页里已经有很多方法了。比如说我想让他弹一个弹一个创建词条的菜单表单出来，弹一个弹窗出来，我已经有个on open了，你看这个是创建词条的这个按钮，我能去点这个按钮和AI去调这个on open，它效果是一样的。

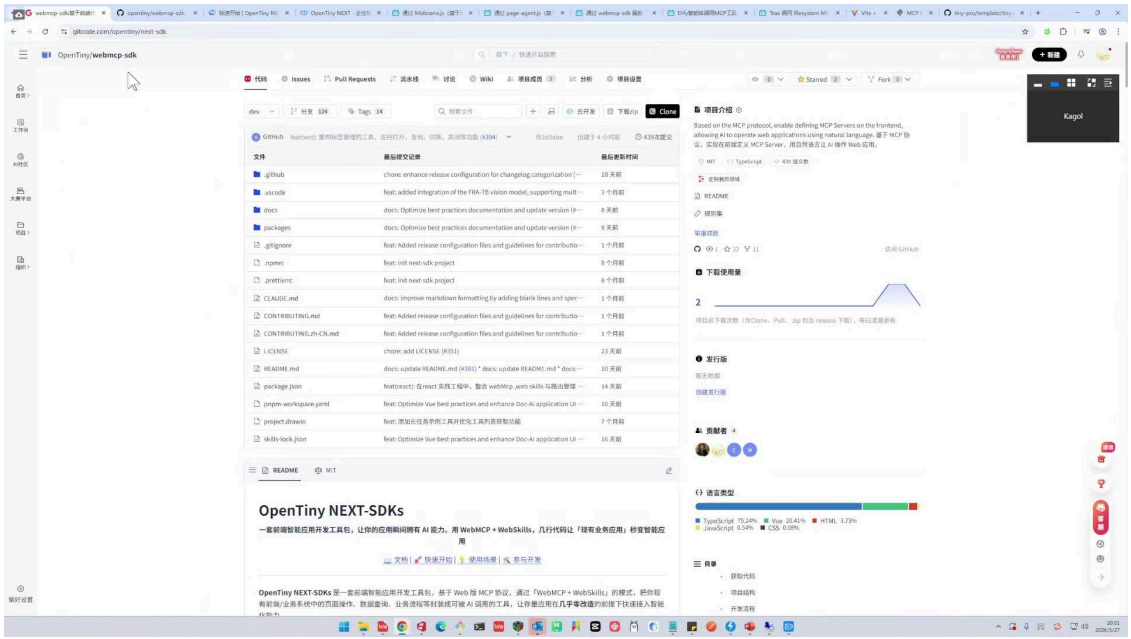


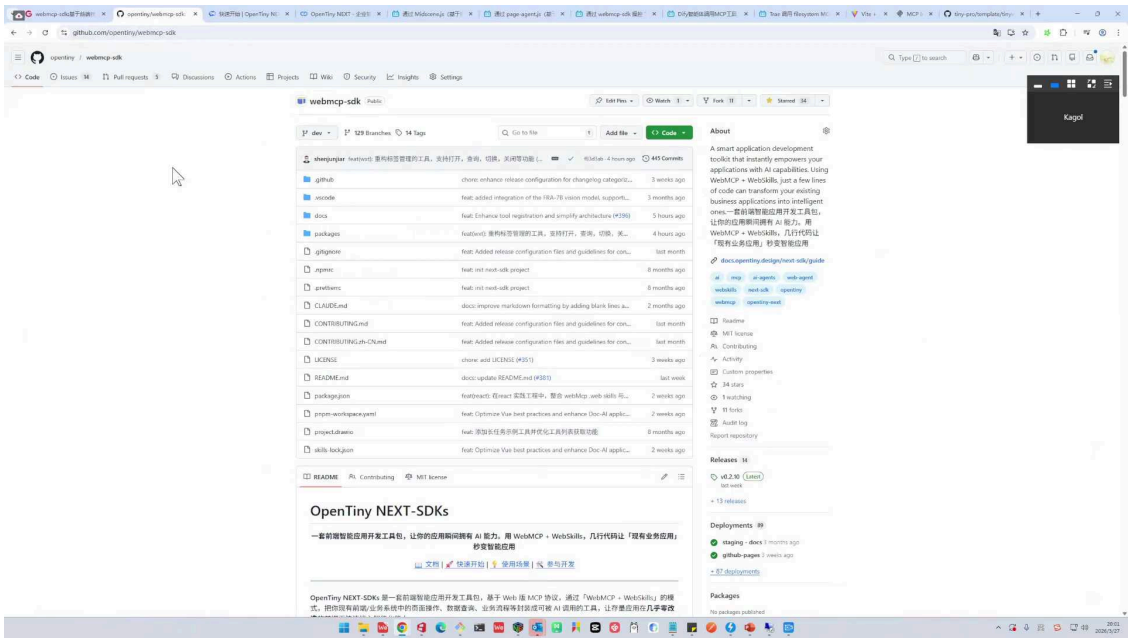
58:16

就是你调这个函数弹框出来，然后再去给这个弹窗里的表单给它赋值。这个表单放组件，他就model里面绑了个local，那你给他logo一个个赋值，赋完值你再去，最终去点确定。At local也是现有的工具，就是我不需要写很多代码，我只要把现有的函数一个拼起来就行了。我能去点确定和AI去调这个at local，它效果是一样的。所以说我觉得他成本肯定是有，但是他带来的收益我觉得是比你付出的成本是多的。

59:03
是否可以基于页面投影出来色差和位置信息做操作？投影出来色差和位置信息，这个色差和位置信息我理解还是基于视觉识别或者基于动的。主要是基于视觉识别和基于动的，它的天然的劣势就是不准确，然后消耗token多，然后非常慢。而且它取决于像视觉识别，他对大模型的要求非常高。就是我们这个工具外部NCB的工具，它对大模型要求非常低。就是你只要能调工具，你能能支持function call，你就能够去操作。只是你强的大模型操作快，弱的大模型操作慢一点，但是也不会慢到哪里去。但是你视觉识别的话，像那个mac点GS它是视觉识别，你可以看他的官网必须要视觉大模型，就非视觉大模型根本就不可用。

01:00:00
目前是不是不是很成熟？我们这个M外部MCBSDK已经从2025年的三月份开始孵化，到现在其实也有快快差不多一年多的时间了。目前的话在公司内部也是有有不少业务已经在用了，所以说目前相对还是比较成熟的。当然这个是需要持续优化的，我们也是个开源项目。如果大家感兴趣也可以去我们的给大家去打开我们的这个项目。我们在这个decode这边也有有做开源，然后所有的代码都是在这里面的。大家如果感兴趣有能力也可以参与共建，可以把它打磨的更好，也欢迎大家给我们点个star就是支持一下。然后我们在这个get hub这边也是做了开源，然后也可以给我们点个支持一下。





01:01:03

底层是什么原理的话，刚刚其实大致讲了一下，就是他他是利用MCP这个协议，把这个MCP就是AI agent的调MCP。他原来是只能调server端的MCP。那现在你把这个MCP搬迁到web里面来，用web MCP的方式让AI去调这个web MCP的工具，从而实现快速的一个操作。那基本上就是这样的原理。

01:01:30

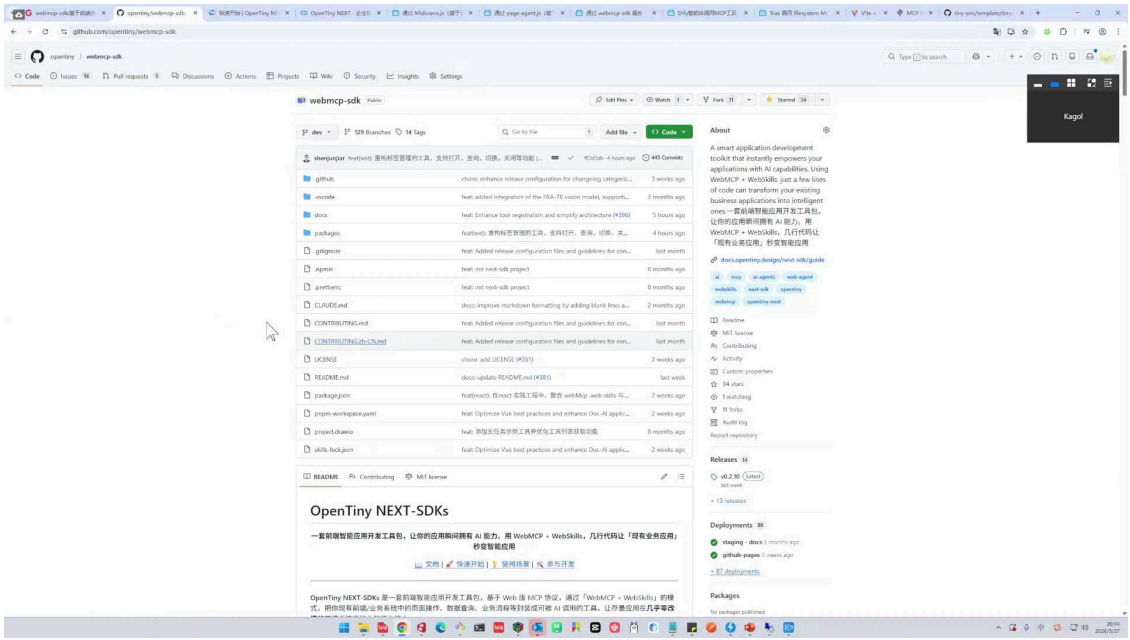
然后具体细节的原理的话，后面我们还有一系列的直播。就是关于我们这个open time next的这个前端智能化解决方案，他到底是怎么回事？整个后面有大概十多期的直播，所以大家可以关注一下，把直播间可以点个关注，就是后续我们可以深入的一个探讨，也可以加我们的这个群。我们也有微信群，就在左下角有个微信群。大家也可以拿出手机去扫一下码，加入我们交流群。后续的话有任何问题，我们很多技术专家都在群里面，都可以给大家做详细的一个解答。

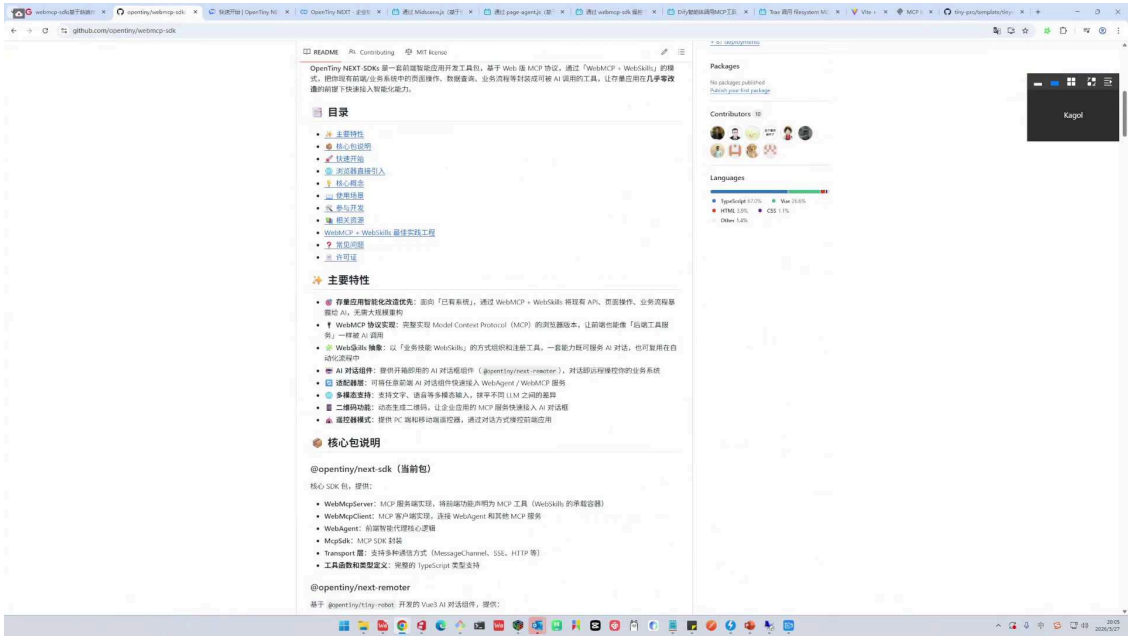
01:02:12

对，这个人说的是对的，就是你到底消耗多少token的话，对这个提示词。这一部分。就你一个工具，你描述的越详细，你肯定AI就要看更多的提示词，就要看更多的信息。但整体来说还是比较消耗，推广还是比较少的。可以部署在内网是可以的，就是可以支持私有化部署，而且可以用自己的私有化这个大模型都是可以支持的。

01:02:57
Web　mcb是基于段或者特定选择器进行操作的。如何应对前端页面的频繁迭代？这个理解可能有点偏差。就是外部MCP它并没有并不是基于段或特定选择器进行操作，它是基于这个MCB工具的。也就是说AI只需要调工具，它不需要真的去操作段。然后操作段或者说执行这个JS这个是你你自己在工具的回调里面写的，它不是AI去执行，你这个不是AI去掉这个状。AI只是去执行你的MCP工具。

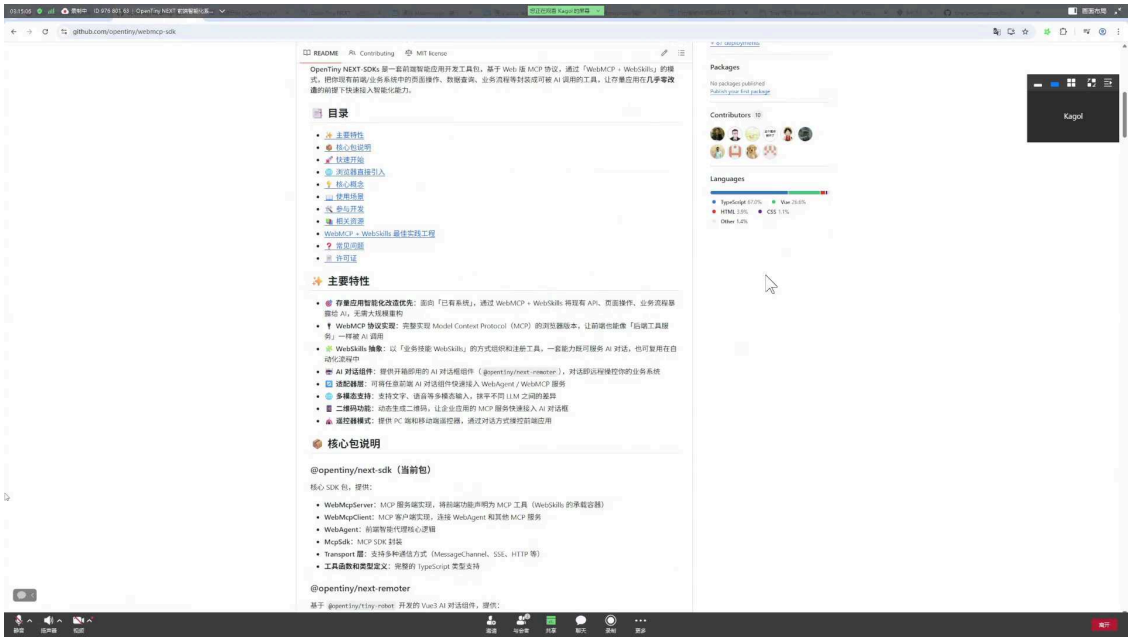
01:03:42
对，这个能理解。对，并不是基于撞的，是一个tour走的function　call，然后暴露给MCV客户端是对的，这个理解是正确的。渐进式加载刚刚提到的web　skill，这个就是实现这个渐进式披露的，就是我不需要把所有的MCP工全部都定在这个APP点view里面，不需要我页面的工具，哪个页面有macd工具，我就定义在那里就行了。我只需要AI他只需要去读这个skin就可以了。他不需要读，每次操作都给你调一大堆的读一大堆的MCB工具的描述，不需要。我给大家去看一下，他这里有个web　scare，就是可以把这个web　scare集成到你的这个项目中来，然后你在外部在里面去做一些相应的描述。然后AI先去看外部，然后再决定他要调什么工具，跳哪个页面。因为你页面不打开，工具不会就是工具不会加载，这个叫渐进式加载和披露，这个是支持的。





01:04:50

你甚至可能首页一开始一个工具都不需要定义，都可以，只要有个scare文件。每个任务都要定义一下。对，就是你只要想要让AI执行的任务，你都要通过工具去给他，通过刘家to去给他定义出来。



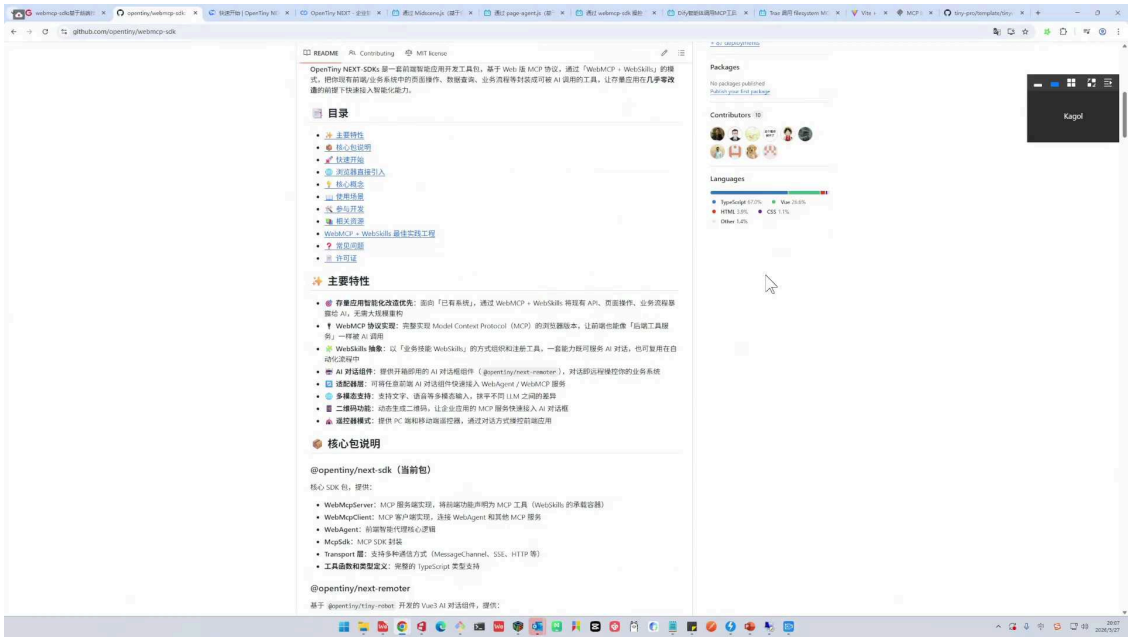
01:05:13

可以用仓颉编程语言实现这个web MCB协议，这个我可能解答不了。收费吗？不收费，这个是开源免费的，每个业务都要写。就是你的业务你想要做智能化改造，让AI去通过调MCP的工具的方式去操控它，你就需要去写这个MCP工具。和单纯调API相比，你也可以让AI郡的直接去调，绕过你的外部页面，直接去调你后端的API，然后去执行这些后台数据库这些操作或者后台接口，这个也是没问题的。但是这个方式也有一个缺陷，就是首先安全性可能没法得到保证，你都不知道AI在后面到底背着你干了些什么事情，你都完全不知道的。所以说你觉得会让人觉得不没那么放心。

01:06:07

第二就是整个过程你是看不到的。就是你刚刚看到外部MCBAI在干嘛，你其实你都可以看得到的，但是后端的这种MCP它你是看不到的，还是在背后操作，操作完了再给你反馈，那工作错了怎么办？就比如说他文件系统，他给你把把根目录给删了，那就损失非常大了，这个事情也不是没有发生过。

01:06:38
不改造现有办法有没有快速识别元素？这个的话就是基于段的方案，就是配置agent，这个是基于段的方案，他就不需要改造现有网站。但是刚刚也看到了，就是配置选A的是我们刚刚演示的三个选手当中的第二个，就是阶段的这个方案配的。他是没有完成这三个任务的。他的第二个任务直接把看板这个副菜单删了，导致他根本没有机会完成后面的任务了。



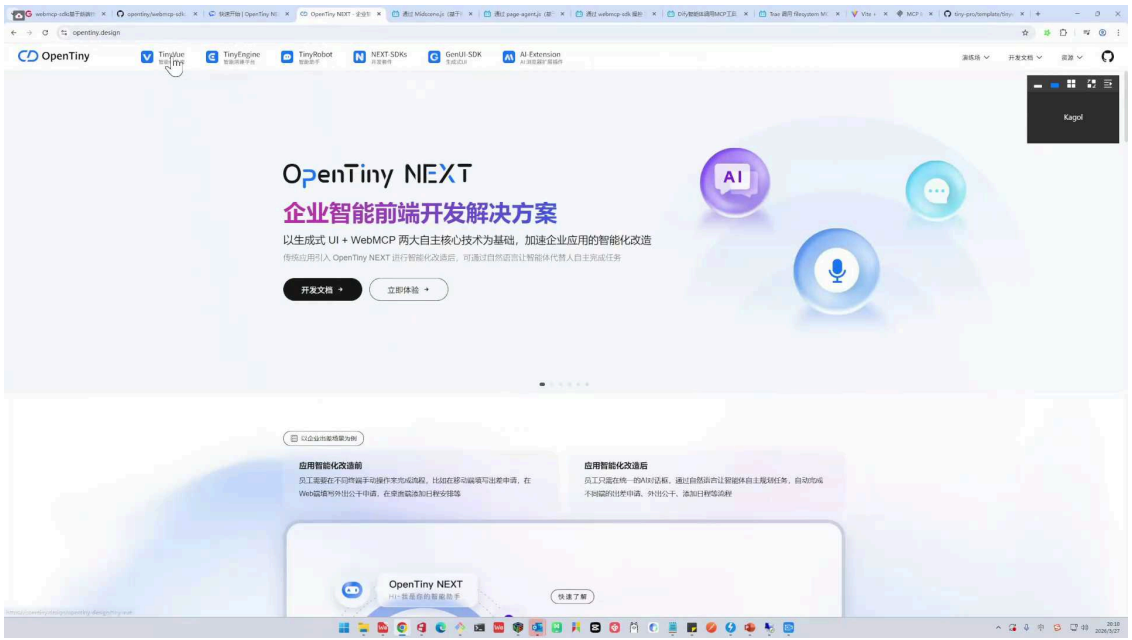
01:07:12
由于时间关系的话，我这边这个问题我就不一定全部解答。但基本上我看了模型不一定百分之百有啥礼品。我想说一点，就是这个方案外观CD它也准确率很高，但不是百分之百，因为大模型它有幻觉，它有可能你描述的没有匹配到MCP工具，或者是匹配错了MCP工具，它调出了工具也会有一些小的偏差。但是我们不能把AI当神来看待，就是你让AI做任何事情，操作网页也好，写代码也好，或者是干什么事情也好，你不能把他当神看待。他只是一个普通的智力比较比较聪明的一个工具，它不是神，所以说幻觉肯定是有的。但是我们可以通过各种方式，比如说scare，描述，或者各种东西去调教它，去优化它，让它能够更好的为我们服务。我觉得这是我对这个AI的一个理解。

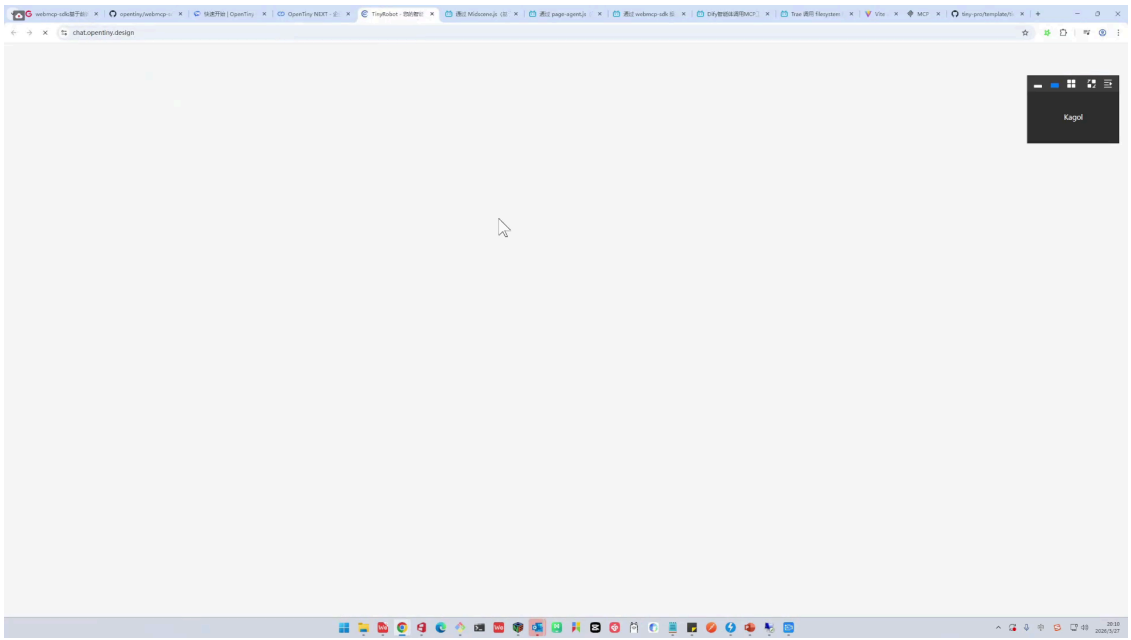
01:08:20
AI也scare也约束不了的行为，有什么好的办法？这个没有特别好的办法，我觉得就是你用提示词用的都约束不了他，那这个就没办法了。这个只能说等着大模型在持续的升级，让它让他理解能力各方面规划变得更厉害。

01:08:43
行，由于时间关系，我们答疑就到这里了。如果。如果大家还有疑问，可以再次提醒一下大家。左下角有一个这个微信交流群的二维码，大家可以扫拿起手机扫一下这个二维码，加入我们的交流群，然后期待和你一起交流AI和外部MCP的最新技术。

01:09:03

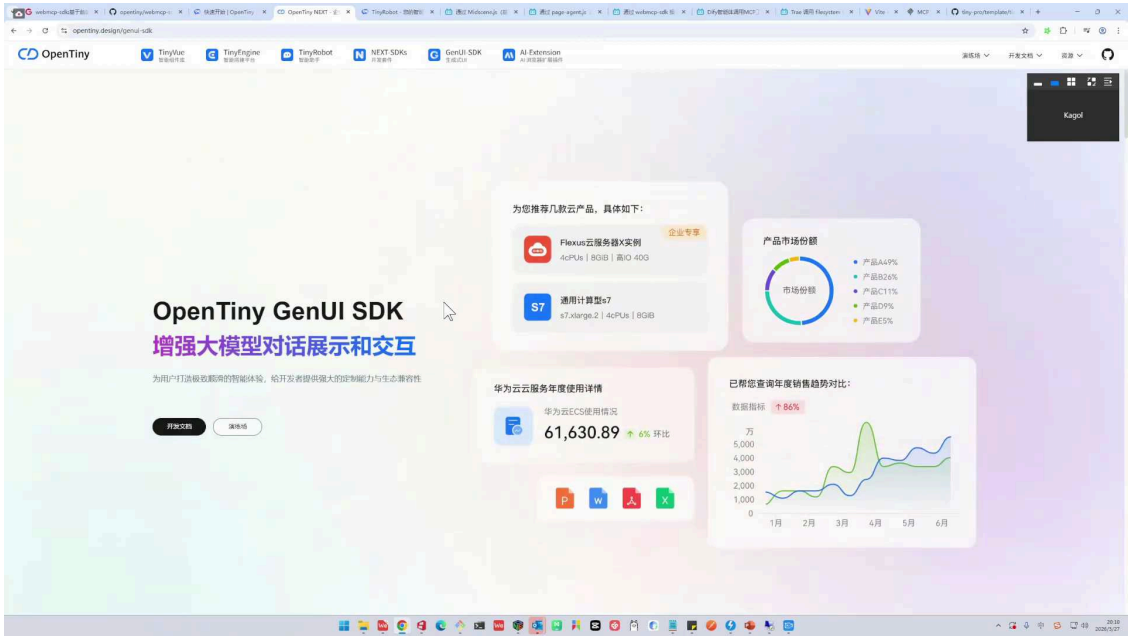
然后目前我们open　time在这个SDK也在这个get　code和getup进行了开源，欢迎大家去点个赞支持一下，非常感谢。然后我们的open time next它是一个全套的一个解决方案，它外部MCPSDK只是其中的一个工具。给大家去看一下我们的一个官网，open开一点，在里面有非常多的这样一个智能化的一个开发的一个产品。包括这个对话助手，就刚刚我们这个遥控器组件就是基于这个对话助手来去去实现的。你可以在这里在这个AI对话助手里面去控制，比如说帮我自增自增级都是没有问题的，这个是OK的。



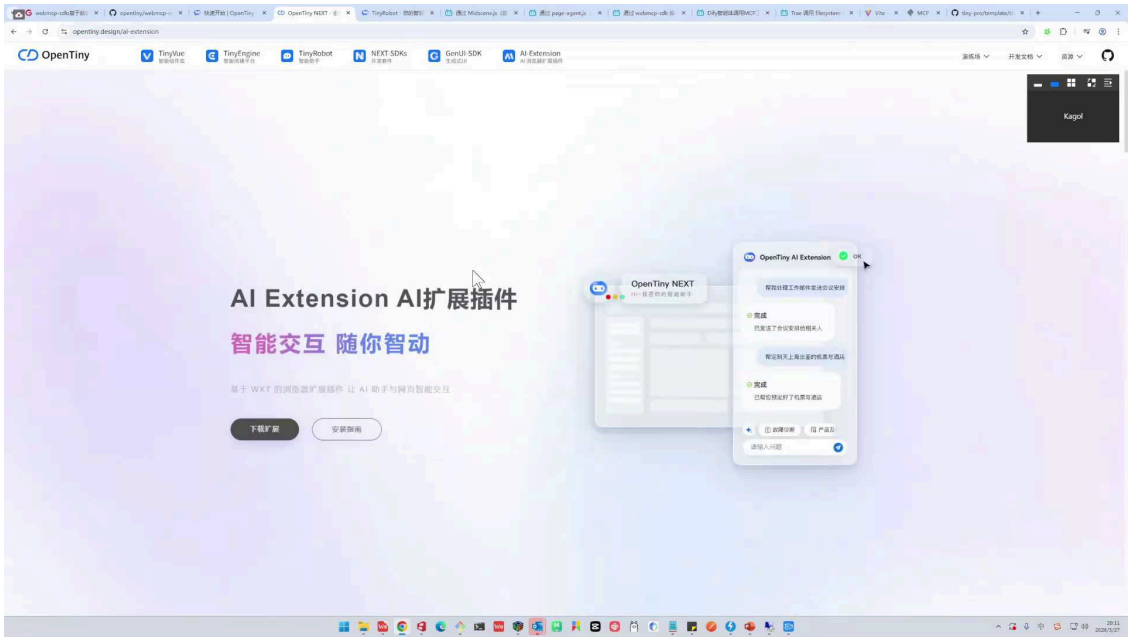


01:09:44

然后另外还有像我们的这个，那也刚刚提到了，还有我们的生成式UI这个就非常厉害了。后面也会给大家去演示这个生成式UI到底是个什么东西，他到底有多强。它能够普通的AI对话框，它只能说你问什么，他给你用文本的方式流式输出，顶多加一个蛋。但是生成式UI可以让你可以用动态让AI动态生成一个非常漂亮的网页去回答你的问题。这个后面大家也可以体验一下，后面我们也有分享去有专家去给大家去介绍这个生成CUI的这个东西。



01:10:23

然后刚才说的AC点GS和page agent点GS我们也有类似的工具叫a extension。这个扩展插件它也可以基于视觉模型和基于那段无障碍信息去对你这个页面进行操控。你不需要改现有页面也可以完成这个操控。但是它的效果就跟max page agent它的这个劣势是一样的，就是没那么准，然后比较慢。但是大家也可以尝试一下，在有些简单的场景下还是可以用的，而且能够给我们做很多的提效。后面我们也有老师会给大家去做这一块的分享。然后我们所有的这些产品都是开源的，然后欢迎大家去使用和参与共

建。接下来我们团队的技术专家也会给大家分享更多关于open tiny next系列产品的解读和实战，欢迎大家持续关注我们。行，我们这次分享就到这里，再见。你不要下面。