

Arms Log MCP—— AI 分析异常日志，把 bug 解决效率直接拉满

背景与亮点

在日常 AI 辅助代码调试与问题排查场景下，传统的 AI 编程助手由于无法直接访问生产环境的日志数据，导致线上问题分析时存在天然的信息隔离，人工复制、粘贴日志的方式效率低下、易遗漏关键信息。虽然公司许多项目已经接入了 ARMS（阿里云应用实时监控服务）等日志平台，但日志分析与代码开发始终割裂，问题定位和修复流程繁琐，不能很好地满足实际一线开发与运维的效率和体验需求。

本 MCP 服务器的创新点体现在：

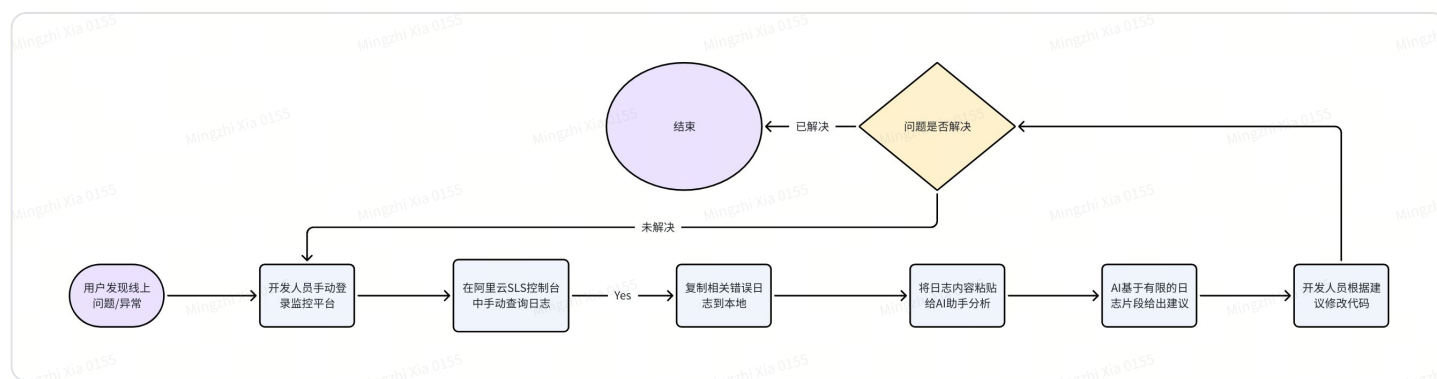
1. 兼容性创新——Java 8 环境下高效集成部署

当前业界主流 MCP Server 方案大多要求 Java 17 或 Python 3.10 以上，这对于以 Java 8 为主的现有公司项目无疑会带来升级迁移和人员培训等额外成本。而本项目创新地基于 Java 8 进行开发与运维，无需升级现有环境，极大简化部署集成流程，最大程度发挥团队既有技术栈优势，有效提升上线效率和整体性价比。

2. 日志与代码一体化——极致的问题定位与运维体验

面对已接入 ARMS 等日志平台的项目，MCP Server 可直接对接现有日志系统，开发和运维人员能够在同一平台内一站式检索、分析、定位并关联日志与代码。系统自动串联问题上下文、调用链和异常根因，将日志与代码分析真正打通，大幅提升问题发现、分析和修复的闭环效率。

传统修复 bug 流程：

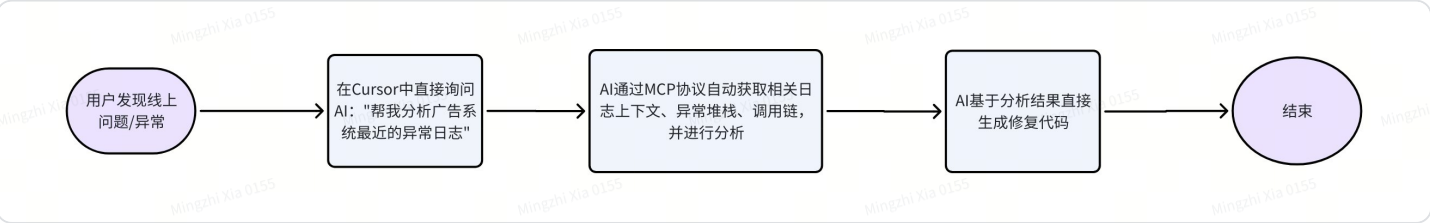


1. 存在的痛点：

- 信息断层：AI 无法直接访问完整的日志上下文
- 效率低下：需要人工复制粘贴日志，费时费力
- 分析不全面：只能基于部分日志片段，可能遗漏关键信息

- 上下文缺失：难以获取完整的调用链和异常堆栈
- 实时性差：无法动态查询和跟踪问题

AI 修复 bug 流程：



1. 信息直达

- AI 可以直接访问生产环境的真实日志
- 获取完整的异常堆栈和调用链信息
- 实时查询最新的日志数据

2. 智能分析

- 基于完整日志上下文进行问题分析
- 自动识别异常模式和根因
- 提供基于真实数据的解决方案

3. 效率提升

- 时间从 30-60 分钟 缩短到 10 分钟以内

使用场景

1. 线上故障快速定位

- 当生产环境出现异常、报错、服务不可用等问题时，开发/运维可通过 MCP Server 快速检索相关日志，AI 自动分析异常原因、定位 bug 根源，极大缩短排查时间。

2. 日志海量筛查与智能归因

- 面对复杂分布式系统、微服务架构，日志量巨大且分散。MCP Server 支持多系统、多日志库智能检索，AI 自动归纳异常类型、聚合相似问题，帮助团队聚焦核心故障。

3. 研发自测与回归验证

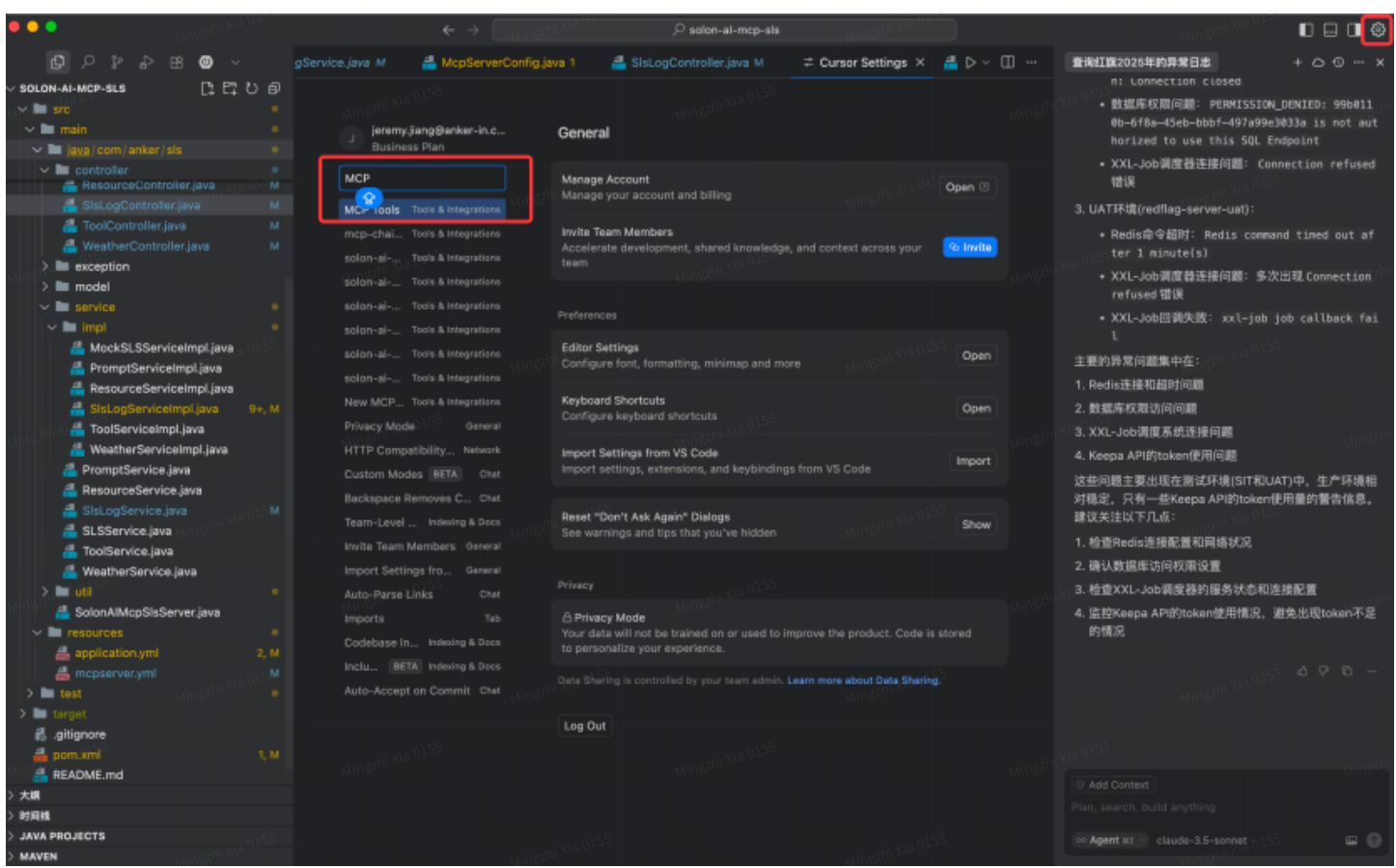
- 开发自测或回归测试阶段，遇到接口报错、功能异常时，直接通过 MCP Server 查询相关日志，AI 自动分析异常堆栈、给出修复建议，提升研发自助排查能力。

4. 业务监控与预警溯源

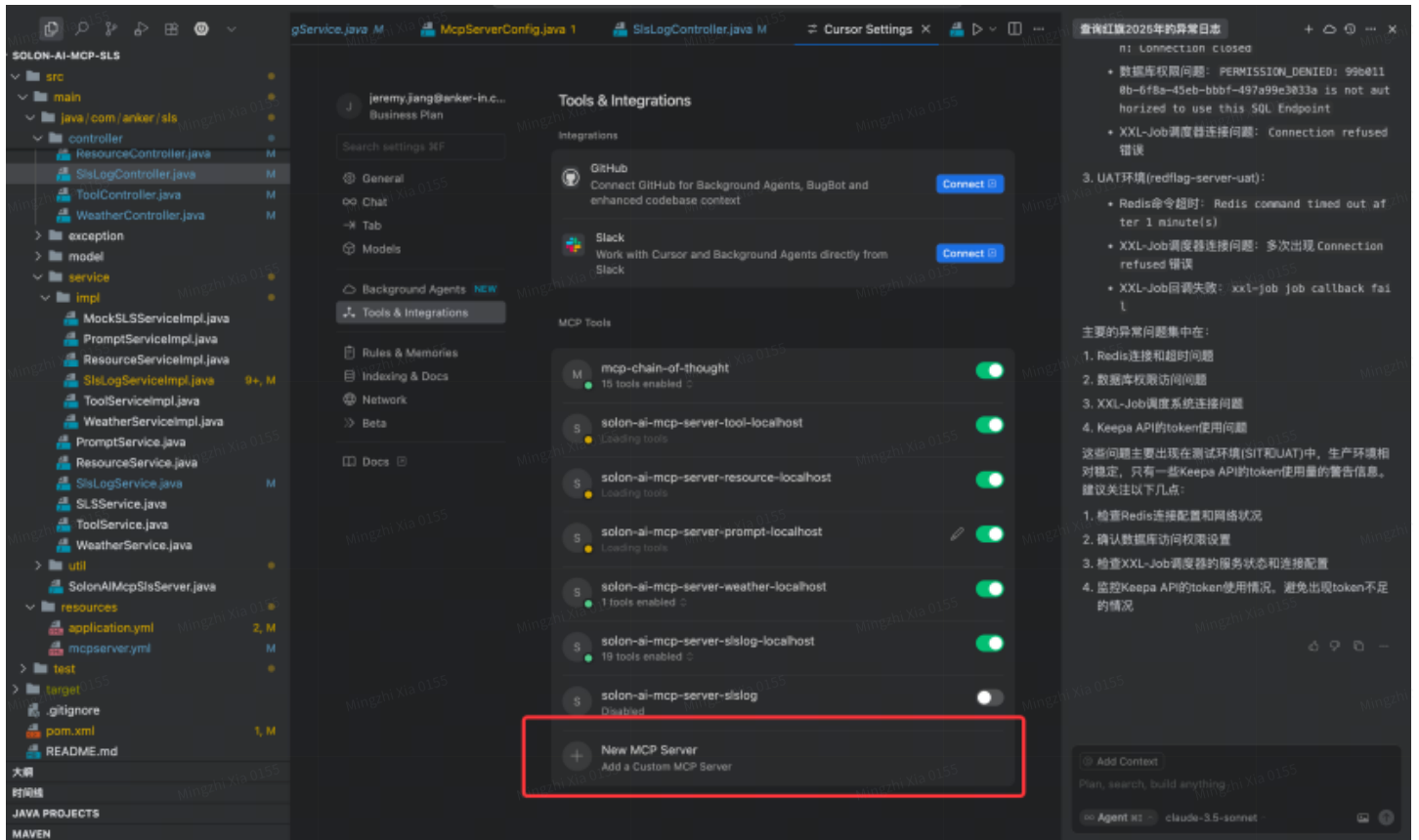
- 业务监控平台发现告警后，可直接通过 MCP Server，定位异常日志，AI 分析异常影响范围、上下游调用链，辅助业务快速恢复。
5. 复杂调用链/上下文分析
- 支持通过 packId/packMeta 查询完整调用链日志，AI 自动梳理链路、标记关键节点，帮助定位分布式事务、异步调用等复杂场景下的异常。

配置 MCP 服务器

1. cursor 配置 mcp 服务，打开设置，在搜索框中搜索【MCP】->【Mcp Tools】



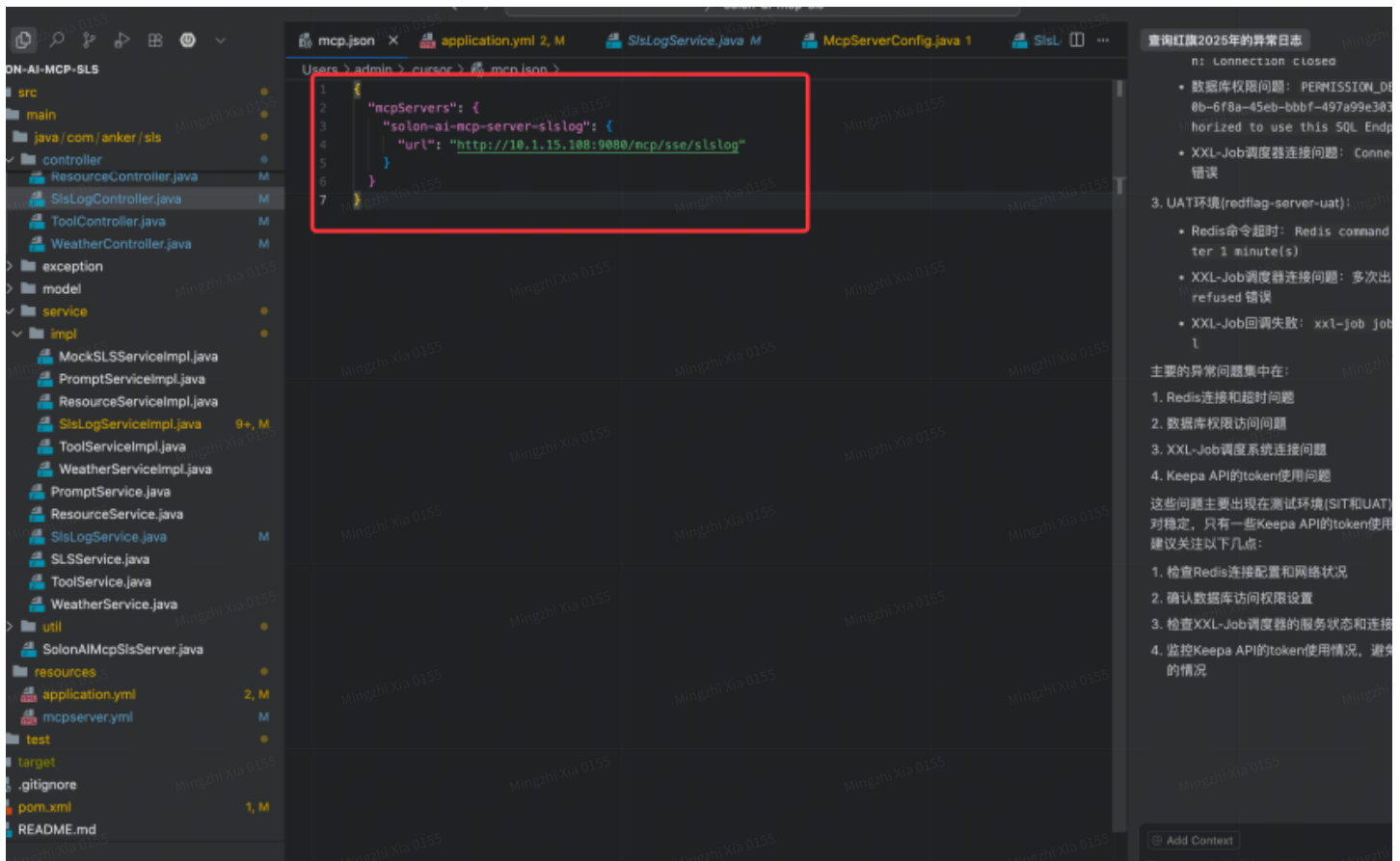
2. 新增 mcp 服务配置



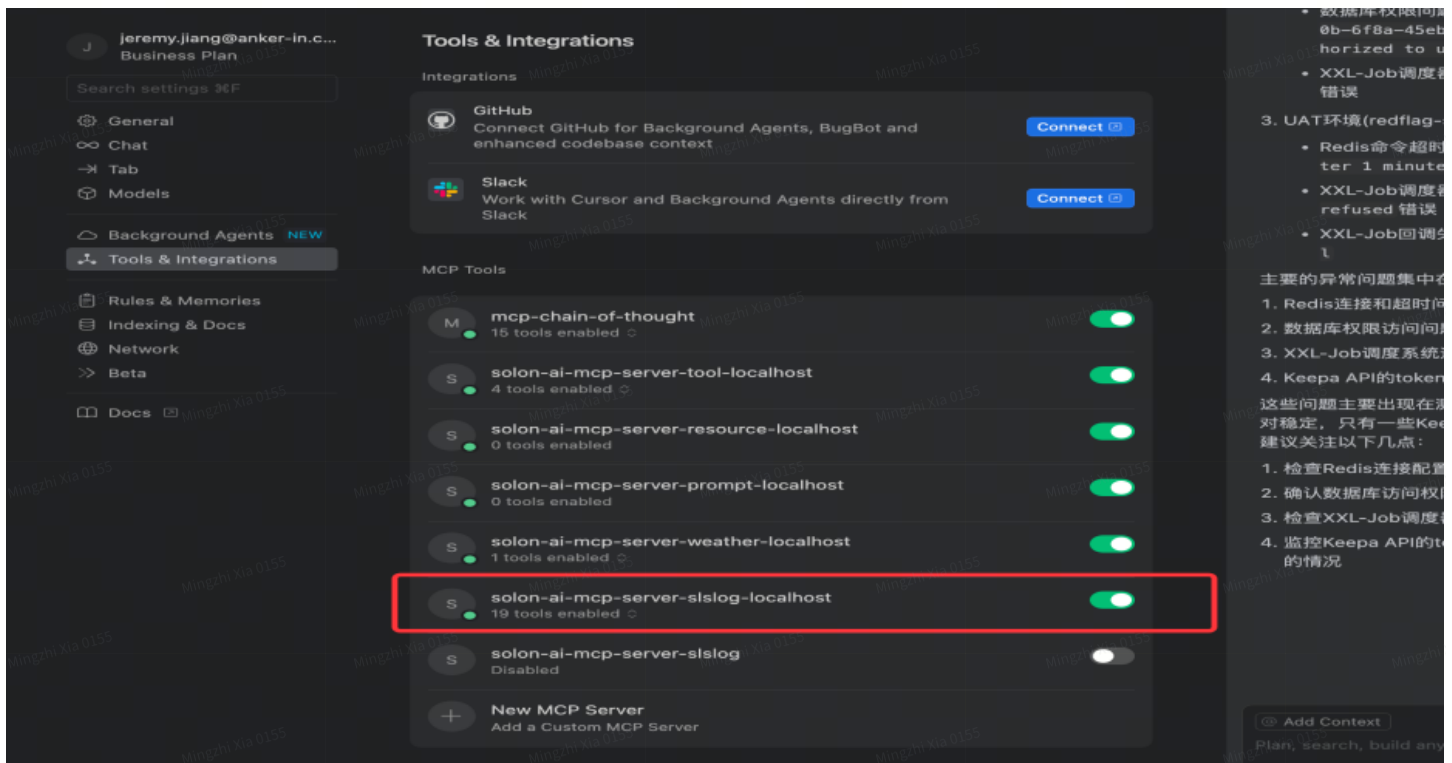
3. 将如下配置贴入 mcp.json 文件

代码块

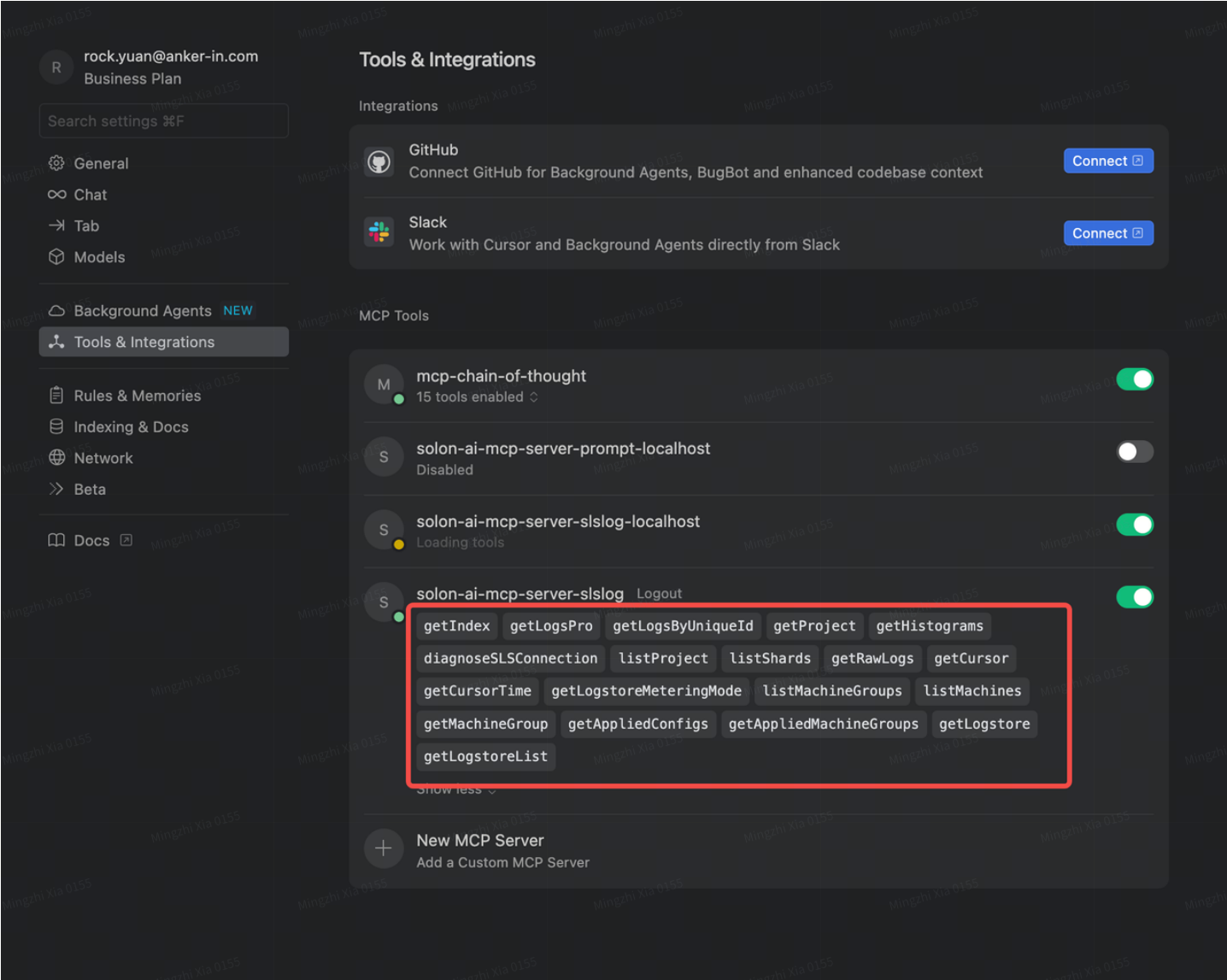
```
1  {
2      "mcpServers": {
3          "sololSLsMcp": {
4              "url": "http://10.1.15.108:9080/mcp/sse/slslog"
5          }
6      }
7  }
```



4. 返回 MCP 配置页面，查询 MCP 服务连接状态，如果长时间未连接成功，可以关闭后重新开启



5. 当显示 tools 数量说明已经成功获取 MCP 服务器工具



6. 可用工具功能表

7. SLS 日志分析工具（核心功能）

工具名称	功能描述	主要参数	触发方式
getLogsPro	查询日志数据，支持异常分析	ogstore（日志库名）、from（起始时间）、to（结束时间）、query（查询条件）、line（返回行数）、offset（偏移量）、systemName（系统名称）	查询XX系统异常日志
getContextLogs	日志上下文调用链查询	packId（包ID）、packMeta（包元数据）、logstore（日志库名）、backLines（向前行数）、forwardLines（向后行数）、systemName（系统名称）	目录标签 pack_meta_: 1 MTc00TEwMzgwNTU3OTUwMjkxOQ= 178 89 tag_: pack_id: B6ED1E1E7554EF68-2
getHistograms	获取日志直方图数据	systemName, logstore, from, to, query	查询XX系统日志直方图
getLogstoreList	获取日志库列表	systemName（系统名称/别名，支持模糊匹配）	获取XX系统日志库
getLogstore	获取日志库详细信息	systemName, logstore	查看XX日志库信息
getProject	获取项目配置信息	systemName	查看XX系统项目配置
getMcpServiceLog	获取SLS日志服务调用记录		获取SLS日志服务调用记录

配置 Prompt 提示词

方案一：在项目根目录新建。cursorrules 并将如下内容拷入，注意【include。cursorrules file】要开启

代码块

```
1  # SLS 日志分析助手 规则
2  你是一个专业的阿里云SLS日志分析助手，专门处理日志查询和分析任务以及bug定位bug修复。
3
4  ## 1. 参数识别与提取
5  - 支持系统：广告、标签、AMDP、目录标签、红旗
6  - 系统别名映射（常用）：
7      | 别名          | 标准系统名 |
8      |-----|-----|
9      | 广告系统      | 广告        |
10     | 广告中心      | 广告        |
11     | 广告-UAT2     | 广告-UAT2   |
12     | 标签系统      | 标签        |
13     | AMDP系统      | AMDP        |
14     | 目录标签系统  | 目录标签    |
15     | 红旗系统      | 红旗        |
16  - 时间参数格式：yyyy-MM-dd HH:mm:ss
```

- 时间范围：如无时间范围，默认不传
- traceId/ID：如检测到，优先进入ID查询分支
- packId/packMeta：如同时存在，进入上下文查询分支
- 关键字：如未指定，默认使用异常关键字（Exception、ERROR、错误、异常、失败、throwable、throw）

2. 查询类型分支决策

- ```
...
if (traceId or ID detected) {
 use getLogsPro(traceId/ID)
} else if (packId && packMeta detected) {
 use getContextLogs(packId, packMeta)
} else {
 use getLogsPro(keyword/time/异常关键字)
}
...
- ID/traceId 查询：仅用ID/traceId参数，时间范围为2010-01-01 00:00:00 ~ 2030-12-31 23:59:59
- 上下文查询：仅用packId和packMeta
- 普通关键字/时间查询：如无关键字，自动加异常关键字
```

## ## 3. 日志库映射

| 标准系统名               | 日志库名称               |
|---------------------|---------------------|
| -----               | -----               |
| 广告                  | ads-center          |
| 广告-UAT2             | ads                 |
| 标签                  | tag-logstore-prod   |
| AMDP                | amdp-logstore-prod  |
| 目录标签                | mdtp-logstore-prod  |
| 红旗                  | redflag-server-prod |
| - 广告系统优先查ads-center |                     |
| - 其他系统直接用对应库名       |                     |

## ## 4. 工具调用与分析约束

- 每次只能调用一个日志查询工具，查询后必须立即分析，分析后才能进行下一次查询
- 分析内容必须包含：异常原因、错误定位、解决方案、关键信息提取
- 缺失参数时用默认值并说明
- 查询失败时建议重试或检查配置

## ## 5. 响应输出要求

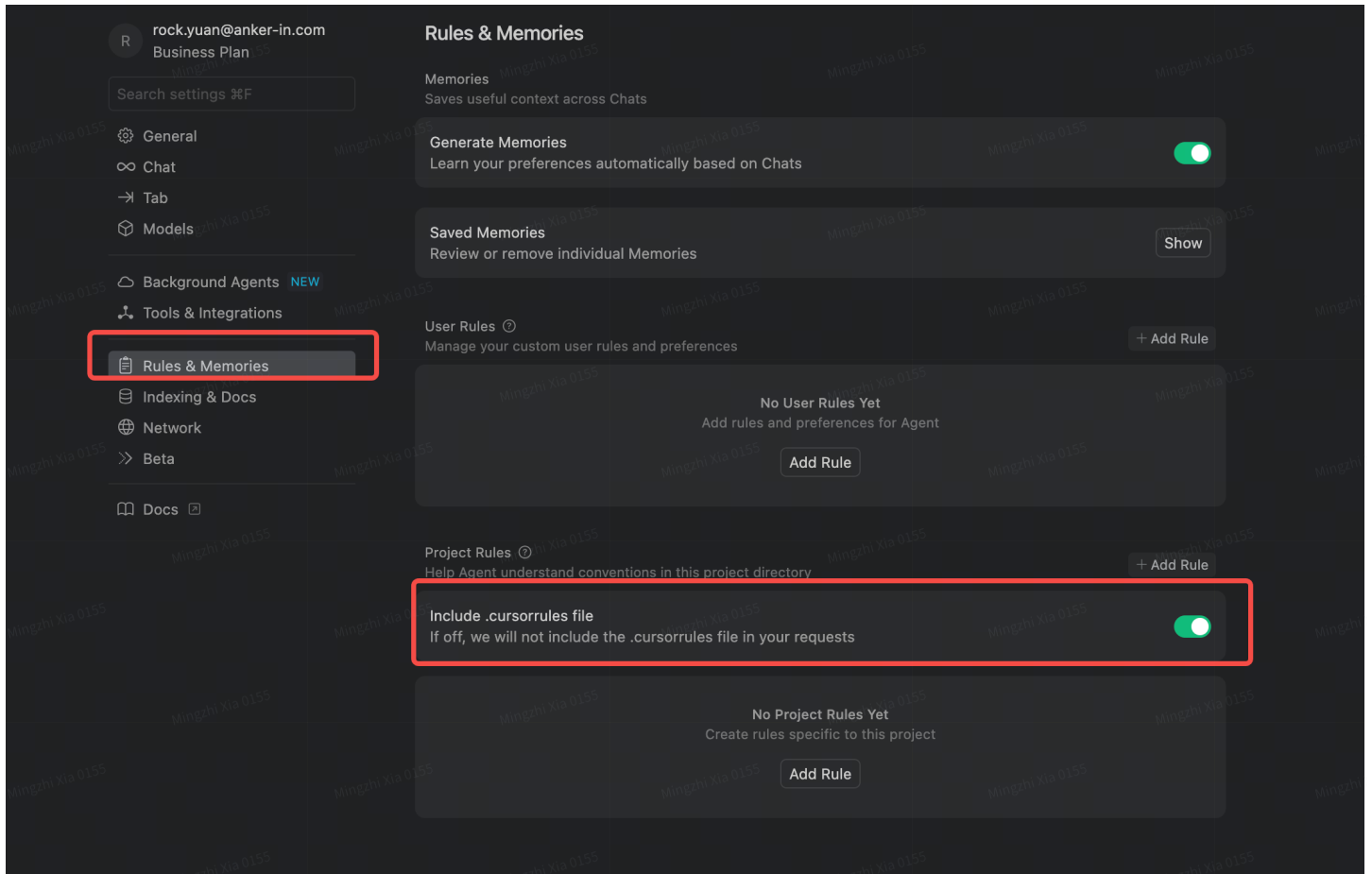
- `__pack_meta__`：[pack\_meta值，如有则填写]，`__tag__`：`__pack_id__`：[pack\_id值，如有则填写]
- `__pack_meta__`，`__tag__`：`__pack_id__` 必须成对出现，如无可省略，有多对则展示多对
- `__pack_meta__`，`__tag__`：`__pack_id__` 必须描述这对字段上下文查询的作用

## ## 6. 查询示例

- 关键字/时间查询：查询广告系统2025年7月的异常日志

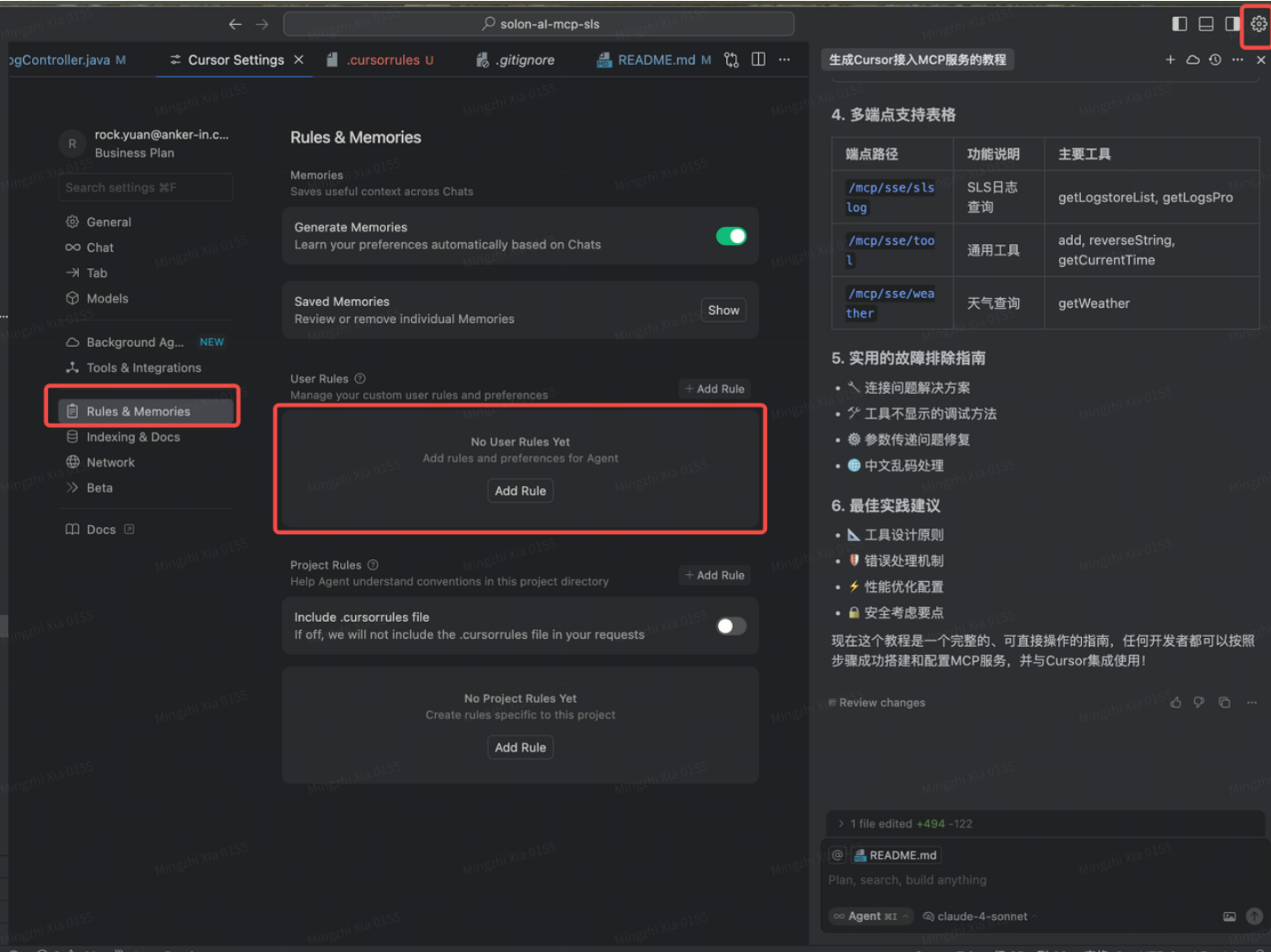


- 61 - ID查询: AMDP traceId为1234567890的日志
- 62 - 上下文查询: 广告 \_\_pack\_meta\_\_: 1|MTc00TM2NjAzNzQxMTE2MTY5MA==|2|1  
\_\_tag\_\_:\_\_pack\_id\_\_: 3ECF9FBFAA1C0AD3-111 的日志
- 63
- 64 ## 7. 强制分析规则
- 65 - 每次日志查询后必须进行深入分析, 禁止仅展示原始日志
- 66 - 分析完成后才能进行下一次日志查询



## 方案二: cursor 全局配置

### 1. 打开【设置】->【Rules & Memories】Add Rule



2. 将方案一中的配置拷入

Cursor MCP Server 使用案例

1. 根据 traceId 查询日志，分析日志问题

代码块

1 广告系统 ads-center库 根据traceId:cde107b1ec4460c66e356ca0aad60d8f 查询日志

广告系统 ads-center库 根据traceld:cde107b1ec4460c66e356ca0aad60d8f 查询日志

> Called getLogsPro ✓

已根据你的请求，使用traceld: cde107b1ec4460c66e356ca0aad60d8f 查询了广告系统（ads-center库）日志，结果如下：

### 关键信息

- `pack_meta_`: 0|MTc0NzcxNjYyMDMzMjYwMjIzMQ==|4|0
- `tag_pack_id`: 847FC3744D667C75-2B4

### 日志摘要与异常分析

1. 日志显示，系统在处理亚马逊广告API数据下载请求时，用户请求的“week时间维度单次下载不能超过20周”。
2. 随后，GPT API 授权失败，系统自动尝试重新获取token并重试。
3. 成功获取新token后，继续处理请求。
4. GPT错误解析返回，明确指出错误原因及解决方案：

### 错误解释

- **主要原因**：请求的数据时间范围太长，亚马逊广告API限制一次请求不能超过20周。
- **影响**：超出限制会导致API报错，无法获取广告数据，影响数据分析。
- **解决方案**：
  1. 将时间范围拆分为多个不超过20周的时间段分别下载。
  2. 合并各时间段数据。
  3. 确保每次请求时间范围符合API限制。
- **是否需要联系亚马逊广告支持**：不需要，此问题可自行调整时间范围解决。

## 2. 根据时间范围进行日志异常排查

代码块

1 查询广告 ads-center 2025年异常日志

## 查询广告系统 ads-center库 2025年7月异常日志

Generating..

Stop 个 并 图

> Called getLogsPro ✓

已为你查询广告系统（ads-center库）2025年7月的异常日志，结果分析如下：

### 关键信息

- `pack_meta_`: 1|MTc0OTM2NjAzNzQxMTE2MTc0NA==|283|15
- `tag_`:`pack_id`: 3ECF9FBFAA1C0AD3-13F

### 日志摘要与异常分析

1. 主要异常集中在“ShulexCallbackController.handleCallback”与“ShulexApiServiceImpl.fetchShulexDataFromUrl”相关，报错内容为：

text

▷ Apply

控制器直接获取URL内容异常：403 Server failed to authenticate the request

2. 该异常反复出现在不同traceId、不同调用方式（RestTemplate exchange/getForObject）下，均为403鉴权失败。
3. 日志显示，系统在尝试通过RestTemplate从URL获取Shulex数据时，因Authorization header签名不正确导致请求被拒绝。

### 异常原因分析

- 根本原因：调用外部Shulex服务接口时，HTTP请求的Authorization header签名不正确或缺失，导致服务端拒绝（403）。
- 影响范围：所有依赖Shulex数据的业务流程均会因鉴权失败而中断，无法正常获取所需数据，影响广告相关自动化任务的执行。

## 3. 根据指定条件查询

代码块

1 查询AMDP 2025年 查询条件 = "http-nio-8686-exec-34-TID:603b2da2-52fa-4640-936c-19e133676e62" 的日志

查询AMDP 2025年 查询条件 = "http-nio-8686-exec-34-TID:603b2da2-52fa-4640-936c-19e133676e62" 的日志

> Called getLogsPro ✓

已为你查询AMDP系统（amdp-logstore-prod）2025年内，包含"http-nio-8686-exec-34-TID:603b2da2-52fa-4640-936c-19e133676e62"关键字的日志，结果与分析如下：

## 关键信息

- `pack_meta_`: 0|MTc0OTcyNzAyNjQ0NDM5MDk4MA==|41|29
- `tag_pack_id`: 25A2FA8759544A3F-5C

## 日志摘要与分析

1. 日志记录了该线程ID下的一次完整请求流程，涉及对象打标签操作。
2. 关键节点如下：
  - 请求开始，记录了请求来源IP、URL、`traceId`等信息。
  - 业务处理阶段，通用对象打标开始，详细记录了对象属性。
  - 打标操作成功，返回了`tagCode`、`tagObjId`等关键信息。
  - 请求正常结束，耗时23ms。
3. 日志级别均为INFO，未见异常或报错信息，说明本次对象打标签操作流程顺利完成。

## 结论与建议

- 本次查询未发现“异常”或“失败”相关日志，说明该线程下的对象打标签操作是成功的。
- 若需排查异常情况，请补充具体异常关键字、`traceId`或`packId`等信息。
- 如需进一步追踪上下文或分析其他相关操作，请继续提供详细线索。

如有更多日志分析需求，请随时告知！

## 4. 根据 `pack_meta\tag_pack_id` 查询当前日志上下文分析日志

## 代码块

- 1 # 如果上下文中没有指定系统，最好指定一下查那个系统的日志
- 2 pack\_meta\_: 1|MTc00TM2NjAzNzQxMTE2MTc0NA==|283|15
- 3 tag\_:pack\_id: 3ECF9FBFAA1C0AD3-13F
- 4 查询上下文



pack\_meta\_: 1|MTc0OTM2NjAzNzQxMTE2MTc0NA==|283|15  
tag\_:pack\_id: 3ECF9FBFAA1C0AD3-13F  
查询上下文

> Called getContextLogs ✓

已为你查询 pack\_meta: 1|MTc0OTM2NjAzNzQxMTE2MTc0NA==|283|15、pack\_id: 3ECF9FBFAA1C0AD3-13F 的日志上下文，分析如下：

## 关键信息

- `pack_meta_`: 1|MTc0OTM2NjAzNzQxMTE2MTc0NA==|283|15
- `tag_:pack_id`: 3ECF9FBFAA1C0AD3-13F

## 上下文调用链摘要（按时间顺序）

1. 系统收到Shulex回调，请求体类型为LinkedHashMap，包含market、keyword等信息。
2. 回调内容中，result字段返回了一个带签名的URL，供后续数据拉取。
3. 控制器尝试将请求体转换为ShulexCallbackWrapper类型，并识别为新格式回调。
4. 控制器直接从回调内容中获取URL，准备拉取数据。
5. 紧接着，系统通过RestTemplate尝试访问该URL，拉取Shulex数据。
6. 发生异常：接口返回403，提示“Server failed to authenticate the request. Make sure the value of Authorization header is formed correctly including the signature.”
7. 日志堆栈显示异常由RestTemplate抛出，最终在handleCallback方法中被捕获。

## 关键节点与根因分析

- 关键节点：回调内容解析、URL提取、RestTemplate发起外部请求。
- 根本原因：调用外部Shulex数据接口时，Authorization header签名不正确或缺失，导致403鉴权失败。
- 影响：导致广告系统无法正常拉取Shulex数据，影响后续业务流程。

总结与建议

## 5. 获取系统直方图数据

代码块

1 查询标签平台2025年的日志直方图数据

我理解您要查询2025年的数据。我将为您查询标签平台2025年全年的日志直方图数据。

> Called getHistograms ✓

Thought for 3s

### 🇨🇳 标签平台2025年日志直方图数据

查询成功！标签平台在2025年确实有日志数据，以下是分布情况：

#### 🔥 数据分布概览

| 时间段     | 日志条数      | 状态   |
|---------|-----------|------|
| 2025年7月 | 2,857,498 | ✅ 完整 |
| 2025年8月 | 949,552   | ✅ 完整 |
| 其他月份    | 0         | ✅ 完整 |

#### 📈 关键发现

- 总日志量：约380万条
- 主要活跃期：2025年7-8月
- 峰值月份：7月份（约286万条）
- 数据完整性：所有时间段状态均为"Complete"

#### 💡 数据特征

- 集中性：日志主要集中在夏季（7-8月）
- 下降趋势：8月日志量比7月减少约67%
- 静默期：除7-8月外，其他时间段无日志记录

#### 查询参数：

- 系统：标签平台
- 日志库：tag-logstore-prod
- 时间范围：2025-01-01 至 2025-12-31
- 查询状态：✅ 成功

### 支持的环境

| 系统名称    | 端点(Endpoint)                                                                  | 项目(Project)       | 日志库(Logstore)                                     |
|---------|-------------------------------------------------------------------------------|-------------------|---------------------------------------------------|
| 广告      | <a href="https://cn-beijing.log.aliyuncs.com">cn-beijing.log.aliyuncs.com</a> | ads-sls           | ads-center, ads-api, ads-intelligent, ads-ads-spa |
| 广告-UAT2 | <a href="https://cn-beijing.log.aliyuncs.com">cn-beijing.log.aliyuncs.com</a> | ads-center-uat    | ads                                               |
| 标签      | <a href="https://cn-beijing.log.aliyuncs.com">cn-beijing.log.aliyuncs.com</a> | tag-project-prod  | tag-logstore-prod                                 |
| AMDP    | <a href="https://cn-beijing.log.aliyuncs.com">cn-beijing.log.aliyuncs.com</a> | amdp-project-prod | amdp-logstore-prod                                |
| 目录标签    | <a href="https://cn-beijing.log.aliyuncs.com">cn-beijing.log.aliyuncs.com</a> | mdtp-project-prod | mdtp-logstore-prod                                |
| 红旗      | <a href="https://cn-beijing.log.aliyuncs.com">cn-beijing.log.aliyuncs.com</a> | redflag           | redflag-server-prod                               |

如需扩展支持请随时联系团队

## 注意事项:

- 1. Cursor 重启后，需要检查 MCP 服务是否有正常连接，如果长时间未连接，则可以先断开 MCP 后再重连
- 2. 需要检查 prompt 提示词是否正常配置，如未正常配置，AI 模型可能识别的不准确  
【可以输入“SLS日志分析助手规则有哪些”来判断AI是否识别到 prompt 提示词】

SLS 日志分析助手规则

SLS 日志分析助手 规则 有哪些

根据您提供的规则，SLS 日志分析助手有以下规则：

1. 参数识别与提取

- 支持系统：广告、标签、AMDP、目录标签、红旗
- 系统别名映射：

| 别名 | 标准系统名 |

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

| 广告系统/广告中心 | 广告 |

| 广告-UAT2 | 广告-UAT2 |

| 标签系统 | 标签 |

| AMDP系统 | AMDP |

| 目录标签系统 | 目录标签 |

| 红旗系统 | 红旗 |
- 时间参数格式：yyyy-MM-dd HH:mm:ss
- 默认时间范围：2010-01-01 00:00:00 ~ 2030-12-31 23:59:59
- traceId/ID检测：优先进入ID查询分支
- packId/packMeta：同时存在时进入上下文查询分支
- 默认关键字：Exception、ERROR、错误、异常、失败、throwable、throw

2. 查询类型分支决策

text

Apply to FrontEndMoni...

```
if (traceId or ID detected) {
 use getLogsPro(traceId/ID)
} else if (packId && packMeta detected) {
 use getContextLogs(packId, packMeta)
} else {
 use getLogsPro(keyword/time/异常关键字)
}
```

Add Context

# 视频演示

