

AI Agent Gateway

📅 2025年4月10日 ⌚ 2 分钟阅读

#AI #Agent Gateway #MCP #A2A

AI Agent Gateway

Agent Gateway

AgentGateway是istio主要服务提供商solo开发的一个专门针对MCP和A2A协议的网关产品。它由rust开发。 solo agentgateway主要解决的问题是：

- 把MCP Server包装成传统的API，然后就可以使用传统的API调用方式来调用MCP Server。
- 支持MCP协议和A2A协议。这个是传统Gateway不能支持的。
- 提供企业级服务，弥补MCP 协议的不足。

Demo： agentgateway docker + mcp/everything server docker

下面的demo演示agentgateway作为mcp-client， everything server作为mcp-server。 client和server都是docker容器， 使用docker-compose来启动。所有步骤如下， 请注意， 大部分步骤都需要修改代码。

修改everything的代码 everything的SSE部分代码已经deprecated了， 但是考虑到agentgateway 的streamhandleHttps部分还没有ready。所以暂时还是使用SSE的方式。主要是把SSE在没有client连接的情况下， server端不会主动推送数据。因为当前这个推送逻辑在没有connection的情况下因为没有try... catch...逻辑， 所以会失败， 并让容器退出。在修改完代码后， 需要重新 build image， 在servers（mcp的github里面）的目录下， 运行下面的命令， 注意， 最后的"."是当前目录的意思。这个命令的写法是由Dockerfile里的"COPY src/everything /app"决定的。

1

```
docker build -f src/everything/Dockerfile -t mcp/everything .
```

在agentgateway的根目录加config.json

config.json的格式参考代码目录crates/agentgateway/proto。 listener.proto 只有一份， target.proto有3份， 分别对应不同协议（mcp, a2a, google?）

目录

文章信息

字数
阅读时间
发布时间
更新时间

标签

#AI #Agent Gateway

mcp/everything里面的sse或streamhandle都用的是3001端口（都是RESTful方式）。stdio方式的mcp server（这里是npx）没有端口，只是agentgateway需要对外暴露以服务端3000而已。

targets部分，我添加了“listeners”部分以便和config.json上面的listeners部分对应起来。

targets部分的docker-everything是用来连接docker-everything的，注意这里的host是“mcp-server”，这个由docker-compose.yml里面的“Services”配置决定，3001在everything的sse.ts或streamhandleHttps.ts里面的代码hardcode的。

```
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48

{
  "type": "static",
  "listeners": [
    {
      "name": "sse",
      "protocol": "MCP",
      "sse": {
        "address": "[::]",
        "port": 3000
      }
    },
    {
      "name": "docker-everything",
      "protocol": "MCP",
      "sse": {
        "address": "[::]",
        "port": 3001
      }
    }
  ],
  "targets": {
    "mcp": [
      {
        "name": "everything",
        "listeners": [
          "sse"
        ],
        "stdio": {
          "cmd": "npx",
          "args": [
            "@modelcontextprotocol/server-
everything"
          ]
        }
      },
      {
        "name": "docker-everything",
        "listeners": [
          "docker-everything"
        ],
        "sse": {
          "host": "[::]",
          "port": 3001,
          "path": "/sse"
        }
      }
    ]
  }
}
```

agentgateway添加docker-compose.yml

mcp-server下面，command部分是用来覆盖缺省的CMD（是stdio），如果要改成MCP的stream的方式的话，把"start:sse"改成"start:streamableHttps"，另外，这里不需要对外暴露port，因为mcp-client只通过mcp-network连接mcp-server，这个在config.json里面配置。

mcp-client下面，volumes使用相对路径，所以要用引号。端口映射部分是为了给外部访问。19000部分，源代码是在127.0.0.1上监听，所以不能对外暴露，我是修改了admin.ts的代码，改成了0.0.0.0。另外3000和3001都是和config.json里面的listeners部分对应的。

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26

```
[
  services:
    mcp-server:
      image: mcp/everything
      container_name: mcp-server-everything
      command: ["npm", "run", "start:sse"]
      networks:
        - mcp-network

    mcp-client:
      image: ghcr.io/agentgateway/agentgateway:v0.4.24-dirty-ext-mcpsvc
      container_name: agentgateway
      ports:
        - "19000:19000"
        - "3000:3000"
        - "3001:3001"
      volumes:
        -
          "/config.json:/root/.config/agentgateway/config.json"
      depends_on:
        - mcp-server
      networks:
        - mcp-network

  networks:
    mcp-network:
      name: mcp-network
      driver: bridge
]
```

MCP inspector测试

运行docker-compose up -d，启动agentgateway和mcp-server。

打开MCP inspector，连接localhost:3000或3001，选择sse协议，点击connect。

在mcp-server的docker容器里面，运行下面的命令，注意这里的"docker-everything"是config.json里面的配置。

1

```
[
  CLIENT_PORT=6274 SERVER_PORT=9000 npx
  @modelcontextprotocol/inspector
]
```

分享这篇
文章



相关文章推荐

Llama 4 模型系列

本文介绍了Llama 4 模型系列详细解读。

CAMEL 工具包

CAMEL Tools

CAMEL工具包是一个模块化框架，旨在通过统一接口扩展AI智能体的能力，使其能够连接外部服务、数据源和计算工具。它提供了多种工具包，涵盖搜索、学术、社交媒体、数据分析、媒体处理、开发、金融和生产力等领域，帮助开发者加速开发、提升可靠性并简化API集成。

CAMEL工具包通过一致的API设计（基于BaseToolkit类）和模型上下文协议（MCP）标准化了工具使用，简化了学习和实施过程。

工具包解决了API集成开销、不一致的接口、网络和错误处理以及维护问题。

主要工具包包括：

网络和搜索工具包：支持多种搜索引擎和知识库，提供实时数据访问。

学术和研究工具包：如arXiv、Google Scholar、

PubMed等，专注于学术文献检索和分析。

社交媒体和通信工具包：
如Twitter、Reddit、LinkedIn等，支持社交媒体数据分析和交互。

数据分析和计算工具包：
如数学、SymPy、NetworkX等，支持数学运算、网络分析和数据处理。

媒体处理工具包：如DALL-E、音频分析、视频分析等，用于图像、音频和视频内容的生成和分析。

开发和编码工具包：如GitHub、终端、代码执行工具包等，支持开发者任务自动化。

金融和商业工具包：如Stripe、OpenBB等，支持支付处理和金融数据分析。

生产力和集成工具包：如MCP、Notion、Excel等，支持项目管理、文档处理和跨平台集成。

CAMEL工具包的优势包括：加速开发、一致接口、可组合性、可靠性与未来兼容性。

不同工具包适用于不同场景，如信息获取、业务优化、创意生成、开发辅助和复杂AI系统。

CAMEL框架通过模块化设计支持工具包的轻松更新和扩展，满足不断变化的市场需求。

1. 网络和搜索类工具包

工具包名称	主要功能	适用场景
搜索工具包	<ul style="list-style-type: none">• Google、Bing、DuckDuckGo等搜索引擎集成• Tavily、Linkup 专业搜索• Wikipedia、Wolfram Alpha知识库访问	<ul style="list-style-type: none">• 事实查询• 最新信息获取• 研究助手开发
浏览器工具包	<ul style="list-style-type: none">• 网页导航• 内容提取• 表单填写• 会话管理	<ul style="list-style-type: none">• 网站数据抓取• 表单自动化• 电商助手开发
天气工具包	<ul style="list-style-type: none">• 全球天气数据获取• 天气预报• 历史记录查询	<ul style="list-style-type: none">• 旅行规划• 物流路线优化• 环境感知服务

2. 学术和研究类工具包

工具包名称	主要功能	适用场景
Anxiv工具包	<ul style="list-style-type: none">• 科学论文搜索• 按关键词/作者/类别检索	<ul style="list-style-type: none">• 研究助手• 预印本跟踪• 文献综述
Google Scholar工具包	<ul style="list-style-type: none">• 学术出版物检索• 引用信息分析• 作者资料查询	<ul style="list-style-type: none">• 跨出版商搜索• 文献计量分析• 研究影响力追踪
PubMed工具包	<ul style="list-style-type: none">• 生物医学文献访问• 临床研究数据库检索	<ul style="list-style-type: none">• 医学研究• 临床决策支持• 制药研究
Semantic Scholar工具包	<ul style="list-style-type: none">• 语义相关性搜索• AI驱动的文獻分析	<ul style="list-style-type: none">• 语义分析• 跨学科研究• 趋势识别

3. 社交媒体和通信类工具包

工具包名称	主要功能	适用场景
Twitter工具包	<ul style="list-style-type: none"> 推文检索 话题跟踪 个人资料分析 	<ul style="list-style-type: none"> 社媒监控 品牌声誉管理 趋势分析
Reddit工具包	<ul style="list-style-type: none"> 帖子检索 评论分析 讨论跟踪 	<ul style="list-style-type: none"> 内容聚合 情感分析 趋势发现
LinkedIn工具包	<ul style="list-style-type: none"> 专业资料检索 公司数据分析 职位信息获取 	<ul style="list-style-type: none"> 招聘助手 职业发展 商业智能
Slack工具包	<ul style="list-style-type: none"> 消息发送 频道管理 对话历史记录 	<ul style="list-style-type: none"> 工作效率工具 团队协作 工作流集成
WhatsApp工具包	<ul style="list-style-type: none"> 消息收发 联系人管理 聊天记录访问 	<ul style="list-style-type: none"> 客服机器人 预约提醒 电商通讯

4. 数据分析和计算
类工具包

工具包名称	主要功能	适用场景
数学工具包	<ul style="list-style-type: none">• 基础到高级运算• 单位转换	<ul style="list-style-type: none">• 金融计算• 工程计算• 数据科学
SymPy工具包	<ul style="list-style-type: none">• 符号数学运算• 微积分计算• 矩阵处理	<ul style="list-style-type: none">• 高级数学教育• 工程研究• 定理证明
NetworkX工具包	<ul style="list-style-type: none">• 图分析• 网络结构创建• 图算法实现	<ul style="list-style-type: none">• 社交网络分析• 路由优化• 推荐系统
Data Commons工具包	<ul style="list-style-type: none">• 公共数据访问• 统计分析• 人口统计	<ul style="list-style-type: none">• 政策分析• 社会经济研究• 公共卫生

5. 媒体处理类工具包

工具包名称	主要功能	适用场景
DALL-E工具包	<ul style="list-style-type: none">• 文本生成图像• 图像修改• 风格控制	<ul style="list-style-type: none">• 创意设计• 营销原型• 概念可视化
音频分析工具包	<ul style="list-style-type: none">• 语音识别• 声音分类• 语音分析	<ul style="list-style-type: none">• 语音助手• 内容审核• 音乐推荐
视频分析工具包	<ul style="list-style-type: none">• 对象检测• 场景分析• 动作识别	<ul style="list-style-type: none">• 内容管理• 安全监控• 运动分析
图像分析工具包	<ul style="list-style-type: none">• 对象检测• 图像分类• OCR识别	<ul style="list-style-type: none">• 文档扫描• 内容过滤• 图像搜索
视频下载工具包	<ul style="list-style-type: none">• 视频检索• 格式转换• 元数据提取	<ul style="list-style-type: none">• 内容存档• 教育培训• 媒体分析

6. 开发和编码类工具包

工具包名称	主要功能	适用场景
GitHub 工具包	<ul style="list-style-type: none">• 代码仓库交互• 提交管理• 问题跟踪	<ul style="list-style-type: none">• 编码助手• 代码分析• 项目管理
终端工具包	<ul style="list-style-type: none">• 系统命令执行• 脚本运行• Shell 交互	<ul style="list-style-type: none">• DevOps 任务• 环境配置• 系统管理
代码执行工具包	<ul style="list-style-type: none">• 多语言代码运行• 沙盒环境支持	<ul style="list-style-type: none">• 编程教学• 代码测试• 算法实验
文件写入工具包	<ul style="list-style-type: none">• 文件创建修改• 权限管理	<ul style="list-style-type: none">• 文档生成• 配置管理• 内容自动化

7. 金融和商业类工具包

工具包名称	主要功能	适用场景
Stripe工具包	<ul style="list-style-type: none">• 支付处理• 订阅管理• 客户数据管理	<ul style="list-style-type: none">• 电商支付• 订阅业务• 财务分析
OpenBB工具包	<ul style="list-style-type: none">• 金融数据分析• 市场可视化	<ul style="list-style-type: none">• 投资咨询• 风险评估• 投资组合追踪
MinerU工具包	<ul style="list-style-type: none">• 文档处理• OCR识别• 表格检测	<ul style="list-style-type: none">• 内容提取• 公式识别• 数据结构化
Dappier工具包	<ul style="list-style-type: none">• 实时数据访问• AI推荐	<ul style="list-style-type: none">• 信息检索• 内容聚合• 数据分析

8. 生产力和集成类

工具包

工具包名称	主要功能	适用场景
MCP工具包	<ul style="list-style-type: none">• 多服务器连接管理• 工具生命周期控制	<ul style="list-style-type: none">• 大规模AI系统• 工作流分解• 协作环境
Notion工具包	<ul style="list-style-type: none">• 页面管理• 数据库处理• 内容上传	<ul style="list-style-type: none">• 知识库构建• 项目管理• 团队协作
Excel工具包	<ul style="list-style-type: none">• 电子表格处理• 格式保留	<ul style="list-style-type: none">• 数据提取• 文档转换• 数据分析
Zapier工具包	<ul style="list-style-type: none">• 自然语言命令• 工作流自动化	<ul style="list-style-type: none">• 流程自动化• 服务集成• 任务执行
Open API工具包	<ul style="list-style-type: none">• API集成• 请求处理	<ul style="list-style-type: none">• 多API管理• 服务代理• API测试
AskNews工具包	<ul style="list-style-type: none">• 新闻聚合• 情感分析	<ul style="list-style-type: none">• 新闻摘要• 媒体监控• 趋势检测
Meshy工具包	<ul style="list-style-type: none">• 3D模型生成• 模型编辑	<ul style="list-style-type: none">• 产品设计• 建筑可视化• 游戏内容
Human工具包	<ul style="list-style-type: none">• 用户输入管理	<ul style="list-style-type: none">• 人机协作• 模型优化

工具包名称	主要功能	适用场景
	• 反馈收集	• 决策验证

这些工具包展现了CAMEL框架强大的生态系统，能够满足从基础开发到高级AI应用的各种需求。每个工具包都经过精心设计，既可以独立使用，也可以组合使用以构建更复杂的应用。

QwQ-32B Qwen推理大模型解读

本文介绍了深度求索（DeepSeek）公司推出的...