

MCP 服务器示例 - 天气查询服务

本项目展示了如何使用 FastMCP 构建一个完整的 MCP (Model Context Protocol) 服务器，并以天气查询服务作为实际案例。

📁 项目结构

```
MCP_Server/
├── intro.md          # 详细的构建指南和核心思路
├── weather_server.py # 天气查询 MCP 服务器
├── requirements.txt   # Python 依赖列表
├── test_server.py     # 服务器测试脚本
├── README.md          # 本文件
└── huggingface_docker/
    ├── Dockerfile      # Docker 镜像构建文件
    ├── requirements.txt # 依赖列表
    ├── weather_server.py # 服务器代码（适配 Docker）
    ├── .dockerignore    # Docker 忽略文件
    └── README.md        # Docker 部署指南
```

🚀 快速开始

1. 安装依赖

```
pip install -r requirements.txt
```

2. 运行服务器

Stdio 模式 (默认, 用于本地开发) :

```
python weather_server.py
```

HTTP 模式 (用于生产环境) :

```
uvicorn weather_server:mcp.http_app --host 0.0.0.0 --port 8000
```

3. 测试服务器

```
python test_server.py
```

📚 文档

- [intro.md](#): 详细的构建指南, 包括:

- FastMCP 介绍
 - 安装步骤
 - 核心思路和架构
 - 开发流程
 - 最佳实践
- [huggingface_docker/README.md](#): HuggingFace Docker 部署指南

🛠 功能特性

提供的工具

1. **get_weather(city: str)**: 获取指定城市的基本天气信息
 - 返回天气状况和温度
2. **get_weather_detailed(city: str)**: 获取指定城市的详细天气信息
 - 包括温度、体感温度、湿度、风速、能见度、气压等

使用示例

```
from hello_agents.tools import MCPTool

# 连接到服务器
weather_tool = MCPTool(
    server_command=["python", "weather_server.py"]
)

# 查询天气
result = weather_tool.run({
    "action": "call_tool",
    "tool_name": "get_weather",
    "arguments": {"city": "北京"}
})
print(result)
```

🌐 部署

本地部署

直接运行 `weather_server.py` 即可。

Docker 部署

参考 `huggingface_docker/` 目录中的示例。

HuggingFace Spaces 部署

1. 将 `huggingface_docker/` 目录中的文件上传到 Space

2. 配置 Space 使用 Docker SDK
3. HuggingFace 会自动构建和部署

详细步骤请参考 [huggingface_docker/README.md](#)。

技术栈

- **FastMCP**: MCP 服务器框架
- **Python 3.10+**: 编程语言
- **requests**: HTTP 请求库
- **uvicorn**: ASGI 服务器 (用于 HTTP 模式)

许可证

MIT License

贡献

欢迎提交 Issue 和 Pull Request!

相关资源

- [FastMCP 官方文档](#)
- [MCP 协议规范](#)
- [HelloAgents 项目文档](#)