

WorldStudioTalk - 我的世界聊天室插件

version 1.0.0

minecraft 1.20.1

java 17

WorldStudioTalk 是一个为我的世界Java版1.20.1服务器设计的高级聊天室插件，支持跨服务器文字聊天互通功能。

功能特性

🎯 核心功能

- 多聊天室管理系统
- 跨服务器聊天互通
- 完整的权限管理
- 自动聊天室分配
- 实时消息同步
- 网络心跳检测
- 配置文件热重载

📋 指令系统

- `/wst help` 或 `/wst h` - 显示帮助信息
- `/wst list` - 查看聊天室列表
- `/wst join <房间名>` - 加入指定聊天室
- `/wst add <房间名>` - 创建新聊天室
- `/wst del <房间名>` - 删除聊天室（仅房主或管理员）
- `/wst exit` - 离开当前聊天室

- `/wst info [房间名]` - 查看聊天室详细信息
- `/wst reload` - 重载配置文件（管理员）
- `/wst status` - 查看插件状态（管理员）

权限节点

- `wst.use` - 使用聊天室基本功能（默认：所有玩家）
- `wst.admin` - 管理聊天室权限（默认：管理员）
- `wst.create` - 创建聊天室权限（默认：所有玩家）
- `wst.delete` - 删除聊天室权限（默认：管理员）

快速开始

环境要求

- Java 17 或更高版本
- Minecraft 1.20.1 服务器
- Spigot/Paper 服务端
- Maven 3.6.0 或更高版本

编译安装

1. 克隆项目

```
bash git clone <项目地址> cd WorldStudioTalk
```

2. 编译插件

```
bash mvn clean package
```

3. 安装插件

```
bash # 将生成的jar文件复制到服务器plugins目录 cp target/  
WorldStudioTalk-1.0.0.jar /服务器路径/plugins/
```

4. 重启服务器

```
bash # 重启Minecraft服务器以加载插件
```

配置说明

插件首次启动后会在 `plugins/WorldStudioTalk/` 目录下生成配置文件：

config.yml 主要配置项

```
# 插件设置
plugin:
    debug: false          # 调试模式
    language: "zh_CN"     # 语言设置

# 聊天室设置
chatroom:
    max-rooms: 50          # 最大聊天室数量
    max-rooms-per-player: 5 # 每个玩家最多创建的聊天室数
    max-members: 100        # 聊天室最大成员数
    max-name-length: 20      # 聊天室名称最大长度
    min-name-length: 3       # 聊天室名称最小长度
    default-room: "全服大厅" # 默认聊天室

# 跨服务器通信设置
network:
    enabled: true           # 是否启用跨服务器通信
    server-id: "server1"    # 服务器唯一ID
    server-name: "主服务器"  # 服务器显示名称
    port: 25580              # 通信端口
    bind-ip: "0.0.0.0"       # 绑定IP
    connections:             # 其他服务器连接配置
        # server2:
        #     host: "192.168.1.100"
        #     port: 25581
        #     name: "生存服务器"
```

跨服务器配置

服务器A配置 (主服务器)

```
network:
  enabled: true
  server-id: "main"
  server-name: "主服务器"
  port: 25580
  bind-ip: "0.0.0.0"
  connections:
    survival:
      host: "192.168.1.100"
      port: 25581
      name: "生存服务器"
```

服务器B配置 (生存服务器)

```
network:
  enabled: true
  server-id: "survival"
  server-name: "生存服务器"
  port: 25581
  bind-ip: "0.0.0.0"
  connections:
    main:
      host: "192.168.1.99"
      port: 25580
      name: "主服务器"
```



使用说明

玩家使用流程

1. 玩家进入服务器

- 自动加入默认聊天室"全服大厅"
- 收到欢迎消息和使用提示

2. 查看聊天室列表

```
/wst list
```

3. 创建聊天室

```
/wst add 我的聊天室
```

4. 加入聊天室

```
/wst join 我的聊天室
```

5. 聊天

- 直接在聊天框输入消息
- 消息会自动发送到当前聊天室
- 支持跨服务器消息同步

6. 离开聊天室

```
/wst exit
```

管理员功能

• 查看插件状态

```
/wst status
```

• 重载配置

```
/wst reload
```

• 删除任意聊天室

```
/wst del 聊天室名称
```



消息格式自定义

在 `config.yml` 中可以自定义各种消息格式：

```
message:  
    chat-format: "&7[&b{room}&7] &f{player}&7: &f{message}"  
    cross-server-format: "&7[&e{server}&7] &7[&b{room}&7] &f{player}  
&7: &f{message}"  
    system-format: "&7[&aWST&7] &f{message}"  
    error-format: "&7[&cWST&7] &c{message}"
```

可用占位符：

- {room} - 聊天室名称
- {player} - 玩家名称
- {message} - 消息内容
- {server} - 服务器名称

API 开发

获取插件实例

```
WorldStudioTalk plugin = WorldStudioTalk.getInstance();
```

聊天室管理

```
ChatRoomManager manager = plugin.getChatRoomManager();  
  
// 创建聊天室  
manager.createRoom("测试房间", "玩家名", false);  
  
// 玩家加入聊天室  
manager.joinRoom(player, "测试房间");  
  
// 发送消息到聊天室  
manager.sendMessageToRoom("测试房间", "玩家名", "Hello World!");
```

网络通信

```
NetworkManager network = plugin.getNetworkManager();  
  
// 广播消息到所有服务器  
network.broadcastMessage("聊天室", "玩家", "消息");  
  
// 获取网络状态  
Map<String, Object> status = network.getNetworkStatus();
```

故障排除

常见问题

1. 插件无法加载

- 检查Java版本是否为17或更高
- 确认服务端版本为1.20.1
- 查看控制台错误信息

2. 跨服务器通信失败

- 检查防火墙设置
- 确认端口配置正确
- 查看网络连接配置

3. 聊天室消息不显示

- 检查玩家权限配置
- 确认聊天室存在
- 查看插件调试日志

调试模式

在 `config.yml` 中启用调试模式：

```
plugin:  
  debug: true
```

启用后插件会输出详细的调试信息到控制台。



更新日志

v1.0.0 (2025-07-15)

- ✨ 初始版本发布
- ✓ 完整的聊天室管理系统
- ✓ 跨服务器通信功能
- ✓ 权限管理系统
- ✓ 配置文件系统
- ✓ 命令自动补全
- ✓ 网络心跳检测

贡献

欢迎提交Issue和Pull Request来帮助改进这个项目！

许可证

本项目采用 MIT 许可证 - 查看 [LICENSE](#) 文件了解详情。

作者

World Studio

- 官网: <https://worldstudio.com>
 - 邮箱: support@worldstudio.com
-

感谢使用 WorldStudioTalk! 🎮✨