

「神奇弹幕」

简介

为什么选择神奇弹幕？

神奇

弹幕

[答谢姬](#)

[弹幕姬](#)

[点歌姬](#)

[直播姬](#)

[录播姬](#)

[回复姬](#)

[视频流](#)

[截图制作](#)

[服务端](#)

[其他](#)

AI功能

[AI对话](#)

[弹幕类型识别](#)

[数据库AI查询](#)

[粉丝档案](#)

[知识卡片](#)

[直播建议](#)

[情绪感知](#)

多平台开发情况

[哔哩哔哩](#)

[抖音](#)

运行与升级

[运行](#)

[升级](#)

[备份](#)

[绿化](#)

[调试](#)

[本地模式](#)

[调试模式](#)

预览截图

说明书

使用技巧

[实时弹幕](#)

[设置账号Cookie](#)

[自定义变量](#)

[语法说明](#)

[欢迎弹幕](#)

[示例：只欢迎戴自己勋章](#)

[示例：指定用户欢迎语](#)

[送礼答谢](#)

[示例：指定礼物专属答谢](#)

[示例：不答谢禁言用户](#)

[关注答谢](#)

[示例：特别关注](#)

[定时任务](#)

[示例：今日是否有大航海](#)

[示例：定时联网](#)

[示例：定时监测屏幕内容](#)

[自动回复](#)

[示例：关键词自动禁言](#)

示例：房管远程弹幕禁言

示例：不要欢迎我

示例：设置自己专属昵称

示例：修改指定用户昵称

浏览器插件

远程控制

弹幕样式

多开与备份

无限多开

自动备份、还原

调试日志

可编程设计

工作流的意义

简单介绍

自定义变量

数据变量

特殊用法

打破“仅直播时发送”

自定义bilibili接口

招呼变量

变量计算

示例：礼物价值

常量

代码展开

示例：知识库列表

函数计算

四则运算

数值修改延迟

语法糖

字符串运算

条件判断

优先级

高优先级被忽略情况

注释

软换行

多条弹幕

触发多段代码

发送选项

与条件的区别

强制房管权限

冷却通道

示例：强制欢迎

等待通道

示例：不连续的定时弹幕

子账号选择

示例：使用子账号

命令操作

批量修改配置

显示值列表

示例：显示所有用户积分

示例：显示所有打卡记录

示例：显示用户总金瓜子

解析JSON：网络数据回调

示例：获取主播信息

关于多线程安全性

简单命令示例

示例：自动打卡

示例：自定义朗读弹幕格式

示例：禁言小游戏
示例：上船自动私信
示例：送礼优先点歌
示例：弹幕切歌
示例：远程开关AI弹幕回复
示例：远程添加违禁词

积分查看

示例：所有用户积分
示例：所有打卡记录

事件动作

主程序事件

示例：远程禁言回复

时间事件

示例：打卡计数
示例：高级打卡

点歌姬事件

示例：点歌提示未带勋章
示例：点歌后立即切歌

弹幕事件

示例：复读机
示例：QQ群推送开播消息
示例：下播自动关机
示例：感谢分享直播间
示例：上船声音提示
示例：上船自动设置房管
示例：成为第x位船长
示例：恭喜勋章升级
示例：天选时刻中奖播报
示例：保存弹幕/送礼记录

大乱斗事件

示例：大乱斗结束前提醒
示例：大乱斗蹭积分卡
示例：大乱斗最佳助攻
示例：大乱斗尊严票
示例：自动开启大乱斗
示例：主播互相串门
示例：大乱斗匹配信息
示例：大乱斗连胜（旧）
示例：大乱斗连胜
示例：对面在线舰长播报

PK对面事件

私信事件

示例：自动回复私信
示例：私信闲聊机器人
示例：解析私信内容
示例：自动生成Excel表单
示例：批量发送Excel信息

原始数据

过滤器事件

示例：点歌过滤
示例：只显示舰长进入
示例：过滤免费礼物
示例：自定义过滤器
示例：自定义词库过滤
示例：自定义正则过滤

录播事件

警告事件

服务端事件

| | |
|---------------|--|
| 示例：远程计数 | |
| 全局快捷键事件 | |
| Cron 定时事件 | |
| 多语言支持 | |
| JS语言 | |
| JS语法示例 | |
| 弹幕接口列表 | |
| 处理配置文件 | |
| 网络连接 | |
| Lua语言 | |
| Lua语法示例 | |
| 处理配置文件 | |
| Python语言 | |
| Python示例 | |
| 处理配置文件 | |
| 使用Python虚拟环境 | |
| 指定命令行exe | |
| 从代码文件导入 | |
| 代码文件示例 | |
| 代码文件中使用变量 | |
| Web开发接口 | |
| 浏览器访问 | |
| Socket通讯 | |
| 主程序发送消息 | |
| 接收主程序消息 | |
| 主程序接收消息 | |
| 通用消息接收事件 | |
| 独立cmd接收事件 | |
| 持久化配置 | |
| 读取主程序信息 | |
| 反向控制主程序 | |
| 便携接口 | |
| 默认WS工具 | |
| 网页程序打包 | |
| QA | |
| 为什么不发送弹幕？ | |
| 调试模式 | |
| 一直在获取房间信息？ | |
| 点歌没有反应 | |
| 实时视频不能播放？ | |
| 讯飞语音没有声音？ | |
| 怎么添加弹幕窗口到OBS？ | |
| 数据重复显示怎么办？ | |
| 怎么提bug？ | |
| 参考资料 | |

「神奇弹幕」

—— 你的智能直播小管家

简介

神奇弹幕以直播间为平台，利用 AI 大模型、网络通信技术、音视频技术、智能控制技术将直播有关的功能集成，构建高效的现代化播出环境与观众互动的管理系统，提升直播亲密性、便利性、艺术性，并提供全方位的信息交互功能。通过网络化综合智能控制和管理，实现“以人为本”的全新直播体验。

整合弹幕姬+答谢姬+点歌姬+回复姬+多种大模型AI+工作流(workflow)，支持弹幕聊天、观众互动与管理、数据统计与分析、自动点歌、私信处理、AI自动闲聊、AI直播建议等。最大的特点是可编程控制，像搭积木一样设置各种互动规则，打造专属直播风格！

本程序以打造自动化直播间为主线，以建设高粘性粉丝团体为目标，保持观众互动，实现粉丝裂变，打造直播高峰，做有温度的直播助手。与此同时，规范观众行为秩序，促进直播产业健康有序发展，营造清朗的网络环境。



```
[!guard% = 0, %anchor_roomid%=%room_id%, %strong_notify% = 0, %mount_roomid% = 0]
[%come_time% = %timestamp% - 10]
欢迎 %ai_name% ---
欢迎 %ai_name% 嗨~
欢迎 %ai_name% 进入美少女的直播间~
欢迎 %ai_name% 一起来看美少女~
欢迎 %ai_name% 来学点知识~
欢迎 %ai_name% 来学点知识~
欢迎 %ai_name% 来学点知识~
%ai_name%, 你来了呀!
%ai_name%, 你来了呀!
%ai_name%, 你来了呀!
%ai_name%, 终于等到你~
%ai_name%~
%ai_name%~
```



神奇弹幕

Bilibili-MagicalDanmaku



弹幕姬+答谢姬+点歌姬+录播姬+回复姬
一个软件，帮你解决一切
可编程、自由拓展

shortcuts

快捷操作，
轻松进行各种常用操作

上下房管、禁言、添加、添加特别关心
设置弹幕样式、为用户添加备注等，
均在 右键菜单 完成





Auto reply

自动回复 解放双手、智能化响应

可根据弹幕发言自动回复，针对高密度提问自动冷却，防止多次重复回应。

可设置 房管 专属指令，让整个管理流程更加便捷

Download

下载

神奇弹幕 的运行需要计算机安装 Visual C++ Runtime 2015 (Win 8+自带)

神奇弹幕 v4.3.4

哔哩哔哩直播弹幕姬+答谢姬+点歌姬+录播姬+回复姬

获取 神奇弹幕

点击此处，立即获取最新的 神奇弹幕 客户端

源码开放: <https://github.com/iwxyi/Bilibili-MagicalDanmaku>

Support

帮助与支持

加入神奇弹幕研讨群，获取实时进度
参与测试，提供反馈



神奇弹幕-主播研讨群

有问题？立即加群联系开发者
群号：1038738410

Notification

通知

未经授权，严禁转载！

- * 点歌服务需要使用自己的账号，部分地区可能不受支持
- * 用户敏感信息等数据仅保留在本地，不会上传到服务器中
- * 数据接收依靠互联网，稳定性取决于运行环境自身的网络质量

© 2020-2021 神奇弹幕 保留所有权利。

为什么选择神奇弹幕？

- ✓ 更高效：自动化处理琐事，让你专心直播内容
- ✓ 更亲密：记住每个老粉的喜好，让互动更有温度
- ✓ 更智能：AI规划你的直播路线，越用越懂你
- ✓ 更自由：支持自定义规则，满足各种创意需求

适合这样的你：

- 想提升直播效率的主播
- 希望增强粉丝粘性的创作者
- 需要管理大型直播间的运营者
- 热爱尝试新科技的直播达人

QQ群：427436529，欢迎交流反馈与研究新功能~

神奇

- 完全支持 AI 的直播工具，智能聊天、数据分析、粉丝档案
- 直播数据全部备份，永久保存到数据库，自定义筛选条件
- 主播秘书，一键上播下播、修改分区，直播数据收集与整理
- 根据约束条件（时间、等级、勋章等），按条件智能欢迎/答谢
- 外语自动翻译、AI闲聊回复、生僻字读音等自动化功能
- 粉丝变化、热门榜、大乱斗、船员变化、新人发言等种种数据
- 大乱斗查看双方串门的粉丝、同步视频PK对面消息、跑骚抓人
- 私信处理、感谢分享、最佳助攻、打卡回复、语音播报等
- 全自动点歌、弹幕切歌、自动暂停其它音乐、排队时长提示
- 点歌姬自动切换网易云/QQ/咪咕/酷狗，支持会员与本地导入
- 完全无人值守，远程控制，开播QQ群播报、自动录播
- 播放各音乐平台会员歌曲，没有版权的歌曲自动更换播放源
- 单个粉丝设置特别关心、强制提醒、专属昵称、永久禁言等
- 查看每名观众进来次数、礼物总计、弹幕记录、判断机器人等
- 弹幕语音播放与回复，百度/讯飞/微软语音等多个接口
- 小窗闲聊、迷你视频、快速截图，上班摸鱼挂后台听直播必备
- 自动签到、参与天选、领取小心心、赠送到期礼物
- 大量可编程运算、网页访问，超高自由度的弹幕机器人
- 游戏设计，比如扣1禁言、关注解禁、送礼物优先点歌等
- 更多神奇的隐藏功能，用户自己开发定制或者导入已有扩展

弹幕

以下仅限本程序在“哔哩哔哩”的功能，其余平台尚未实现全部功能。

答谢姬

- 自动欢迎进入
- 自动感谢送礼
- 实时感谢关注
- 冷却避免刷屏
- 合并礼物连击
- 设置礼物别名
- 智能昵称简化
- 目标强制欢迎
- 舰长单独欢迎
- 自动语音播报
- 个人专属昵称

- 开播下播提醒
- 可编程变量集
- 动态条件运算
- 黑白名单筛选
- 支持远程控制
- 支持语音朗读
- 礼物总结答谢
- 答谢分享直播间
- 答谢PK最佳助攻
- PK对面信息播报
- PK在线舰长播报
- PK连胜数量播报

弹幕姬

- 实时显示弹幕
- 小窗弹幕聊天
- 定时弹幕任务
- 简约聊天模式
- 点歌自动复制
- 查看点歌历史
- 保存弹幕历史
- 每日数据统计
- 外语自动翻译
- 智能沙雕回复
- 特别关心高亮
- 查看粉丝牌子
- 查看礼物价值
- 查看昵称拼音
- 粉丝变化提示
- 房管一键禁言
- 新人发言提示
- 弹幕自由复制
- 定时连接后台
- 喷子自动拉黑
- 新人快速禁言
- 跳转用户主页
- 用户累计数据
- 用户弹幕历史
- PK 历史次数
- 快速百度搜索
- 大乱斗的串门
- 对面消息同步
- 机器小号判断
- 弹幕语音朗读
- 支持讯飞语音
- 支持语音接口
- 弹幕全屏滚动
- 背景图片轮播
- 自动切换勋章
- 发送对面弹幕
- 显示船员数量

- 屏蔽用户进入
- 屏幕天选弹幕
- 发弹幕表情包

点歌姬

- 音源自动切换
- 播放会员歌曲
- 网易云ID点歌
- 点歌自动播放
- 智能匹配歌曲
- 暂停其余音乐
- 点歌历史记录
- 自由显示格式
- 自动点歌队列
- 点歌防止刷屏
- 关键词黑名单
- 伴奏优先播放
- 预计时长回复
- 支持弹幕切歌
- 送礼立即播放
- 背景封面模糊
- 提取主题色调
- 查看桌面歌词
- 收藏夹与历史
- 一键导入歌单
- 导入本地音乐

直播姬

- 修改房间标题
- 一键开播下播
- 切换直播分区
- 直播数据展示
- 实时人气查看
- 每日平均人气
- 一键任命房管

录播姬

- 自动开始录播
- 分割文件大小
- 断联自动重录

回复姬

- 自定义关键词
- 筛选用户条件
- 多条随机回复
- 支持编程变量
- 数量不做限制
- 用户弹幕打卡
- 远程控制程序

- 修改专属昵称
- 弹幕文字替换
- 每日天气预报
- 解析用户私信

视频流

- 匿名查看直播
- 预先连续截图
- 预览PK对面
- 避免PK黑屏
- 自动调整比例

截图制作

- 快速预先截图
- 多图片管理器
- 好图一键提取
- 截图批量裁剪
- 分解GIF动图
- 制作GIF动图
- 制作AVI视频
- 制作字符画图

服务端

- 直播姬可访问
- 无限扩展可能
- 支持B站接口

其他

- 开启大乱斗
- 全局快捷键
- 保存头像封面
- 配置导入导出
- 数据自动备份
- 投稿歌词字幕
- 自动切换勋章
- 每日自动签到
- 自动参与天选
- 赠送到期礼物
- 直播状态查询
- 直播匿名跑骚
- 捕捉黑听舰长

AI功能

让直播更懂你，让互动更智能。

AI对话

- 支持 GPT/DeepSeek 等主流大模型自由切换
- 智能生成自然流畅的回复
- 可精准控制回复时机和频率

弹幕类型识别

- 深度理解弹幕语义和意图
- 自动分类：闲聊/翻译/问答/感谢等
- 智能执行对应操作：自动翻译/点歌/提醒等
- 突破传统规则限制，精准识别各种表达方式

比如自动翻译外文、违禁词自动禁言、自动点歌等。有别于基于表达式的机器人，这是大模型识别，不固定范式，解决各种奇葩弹幕。

数据库AI查询

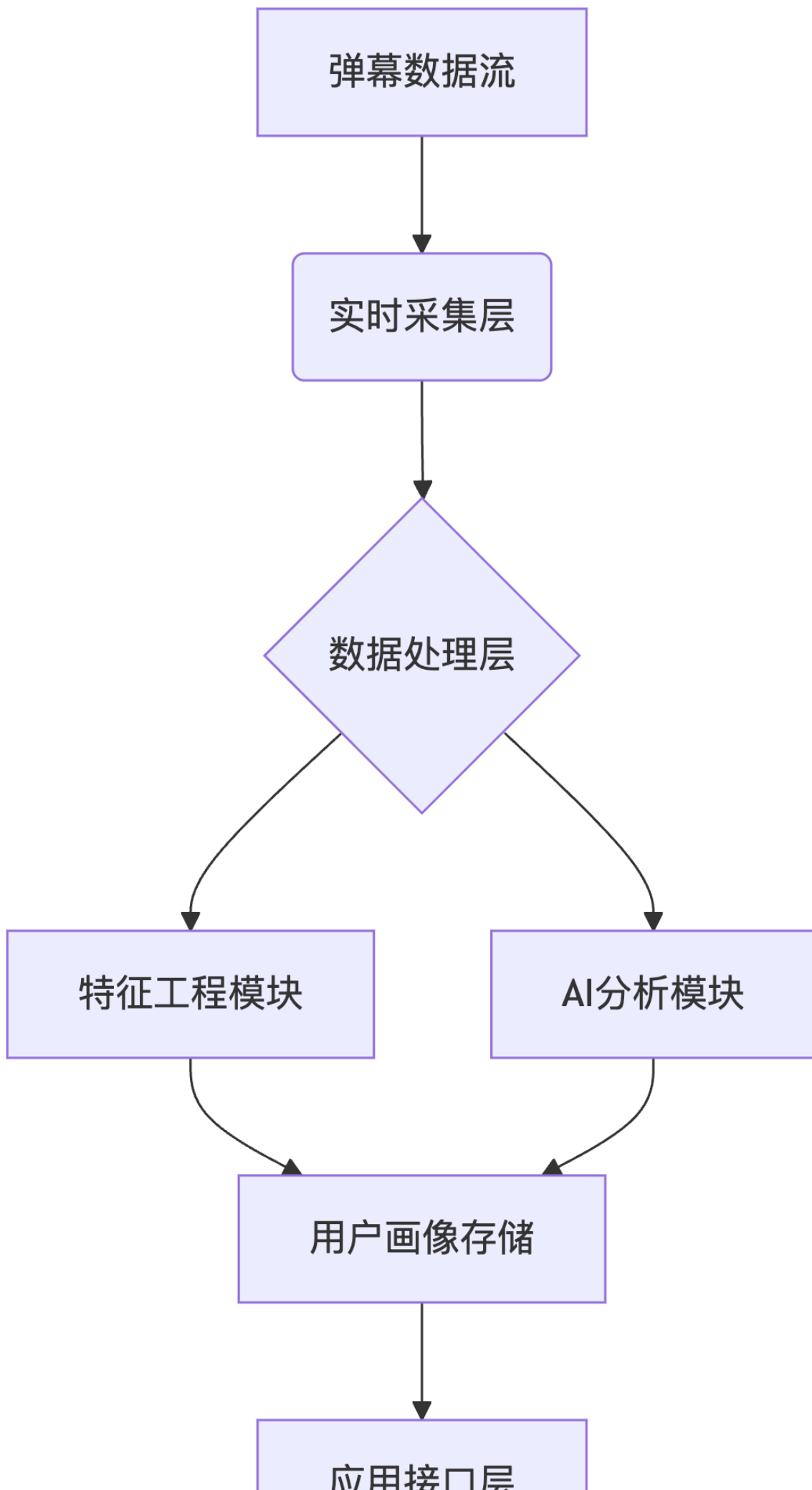
自然语言查询直播数据，例如“查询近30天送礼TOP10观众”、“进入直播间超过10次且送礼金额大于100元的观众”，让数据查询像聊天一样简单。

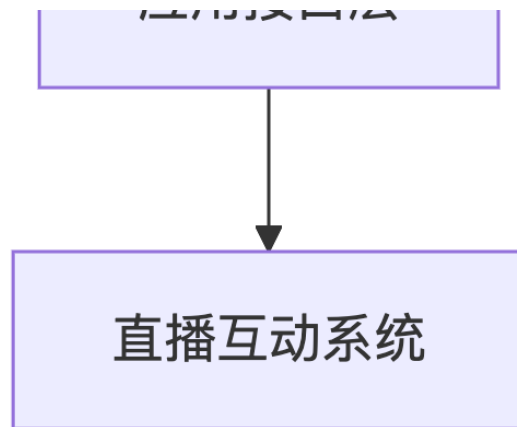
粉丝档案

主播的“小本本”，记录每一位活跃观众在直播间的喜好、性格和互动习惯。

比如：喜欢聊什么、性格特点、消费偏好、活跃时间等，让观众有更懂你的主播，获取专属互动体验。

“让每一条弹幕，都被真正看见，并且记住。”





知识卡片

实时解析弹幕中的专业内容，自动生成简明易懂的知识卡片。
既可主播查看，也可分享给观众，让直播内容更专业、更有深度。

直播建议

基于实时数据智能分析，提供：互动时机/内容调整/观众关怀等建议。
例如“现在可以带货了”、“深夜要关注某位观众的健康，错别字太多了”、“让某某早点睡，明天还要送小孩上学”等等。
让直播决策更科学、更贴心。

情绪感知

实时分析弹幕情感倾向，自动分析观众情感，预警高频负面情绪（如“无聊”出现5次/分钟）。

多平台开发情况

哔哩哔哩

本程序以“哔哩哔哩”平台为主，所有功能基本都可使用。
直播的各接口更新频繁，部分功能随时可能失效，请注意保持更新，

抖音

目前已实现的功能：

- Cookie登录、浏览器登录
- 接收观众弹幕、进房、关注、点赞、送礼

注意以下**核心功能尚未实现**：

- 发送弹幕
- 主播/管理员操作
- 录播、直播画面
- “事件动作”中与该平台相关的都无法获取数据，但可响应

运行与升级

运行

绿色版，免安装，双击 `神奇弹幕.exe` 打开。

退出从屏幕右下角托盘的右键菜单中退出。

升级

下载新版的zip文件，解压后将里面的文件复制，覆盖旧版的文件。

备份

复制整个安装文件夹即可，支持任意多开。

绿化

默认就是绿色版，无限多开，可放在U盘等移动介质上。

删掉程序同一目录下的“green_version”文件，即变成安装版，不同位置的神奇弹幕使用同一套数据。

调试

本地模式

开启后，对账号的修改操作都会变为“展示文字”的形式显示在弹幕姬中，而不进行实际操作。例如发送弹幕、拉黑等等。

调试模式

开启后，将详细显示一些逻辑上的判断点，例如冷却中、空弹幕等等，便于代码问题的查找。

在向开发者反馈问题前，强烈建议按照论坛中的一些流程排查一遍，大部分情况在论坛中都有人提出。

预览截图



14798635

可爱

尊享版



翊Yoko

未开播

这里习习♥ 希望你开心快乐且自由

cos

lolita

jk

你好 我是习习

感谢你在人中发现了我~

Cos/Lolita/制服/汉服 四坑吃土少女



98 85158 211

舰长

粉丝

粉丝团



0 0 0

热门榜 人气值 弹幕



懒一夕智能科技官方

状态: 已连接

637

发送弹幕

广告位招租

召唤弹幕姬



显示时长

弹幕超时300秒



提示超时120秒



☒ 保存弹幕记录

弹幕发送

最长20字



☒ 自动调整字数

☒ 发送失败重试

智能回复

☐ AI回复参考

☐ 发送AI弹幕

允许禁言

各种禁言功能的总开关，仅限房管与主播使用

☒ 启用禁言

大乱斗串门

观众串门报告、显示对面直播间的弹幕

☒ 大乱斗串门小报告

☐ PK同步消息

发言提示

☐ 外语自动翻译

☒ 新人发言提示

☒ 机器人判断(仅关注)

大乱斗偷塔

☐ 大乱斗自动偷塔

上限 ☒ 自动调整

偷塔时间

黑名单房间

开播问候

☒ 启用

上播 欢迎我们的美女主播~

下播 留空则不发送

☐ 自动更换勋章

鼠标穿透

针对游戏主播，避免弹幕姬挡住游戏界面

定时连接

☐ 启用

☐ 一整天

开播21点



下播1点



自动操作

☒ 参与天选之子/节奏风暴

☒ 每日自动签到

☒ 挂机领小心心

☐ 赠送到期礼物

监听

☒ 仅直播时回复

☐ 监听勋章升级

☒ 启用过滤器

每日统计

☒ 每日数据统计

今日

导出

礼物历史

☒ 记录礼物历史

☒ 记录每次上船

☒ 记录每月船员

录播姬

☐ 原画录播

自动分割每30分钟



调试

请输入消息内容

发送弹幕

...

欢迎

送礼

关注

☒ 欢迎进入☒ 文字 ☐ 语音 冷却10秒

```
[%guard% = 0, %anchor_roomid%!=%room_id%, %strong_notify%=0, "%{mount_%uid%}"=]"**
[%come_time%>%timestamp%-10]*
欢迎 %ai_name% ~~~
欢迎 %ai_name% 哦~
欢迎 %ai_name% 进入美少女的直播间~
欢迎 %ai_name% 一起来看美少女~
欢迎 %ai_name% 来看美少女~
欢迎 %ai_name% 来看习习~
欢迎 %ai_name% 光临~
%ai_name%, 你来了呀!
%ai_name%, 你来了呦!
欢迎 %ai_name%, 你终于来啦~
欢迎 %ai_name%, 终于等到你~
嗨, %ai_name%~
哈喽, %ai_name%~
%ai_name%, 哈喽~
[%medal_level% > 20]欢迎 %ai_name%, 请多多关照~
欢迎 %ai_name%, 捕捉一只小可爱~
欢迎小可爱 %ai_name% ~
[0]%ai_name%, %all_greet%~
[0]%all_greet%, %ai_name% ~
[0]%ai_name%, %greet%%tone/punc%
[0]%greet%%tone%, %ai_name%%punc%
[%pk_opposite%]***cd10:2)%ai_name%, 欢迎串门哦~
[%pk_opposite%]***cd10:2)%ai_name%, 欢迎来串门~
[%pk_opposite%]***cd10:2)欢迎%ai_name%串门~
[%pk_opposite%]***cd10:2)欢迎对面的%ai_name%来看习习~
[%pk_opposite%]***cd10:2)谢谢%ai_name%百忙之中串门看习习~
[%pk_opposite%,%uid%=%pk_uid%]****cd10:2)谢谢主播%ai_name%百忙之中来串门~
[%guard%]*(cd8:4)欢迎%guard_name% %ai_name% 光临!
[%guard%]*(cd8:4)欢迎%guard_name% %ai_name% 大驾光临!
[%guard%]*(cd8:4)欢迎%guard_name% %ai_name% 回家~
[%guard%]*(cd8:4)欢迎%guard_name% %ai_name% 归来~
[%pk_view_return%]***cd10:1)欢迎%ai_name%串门回来~
[%pk_view_return%]***cd10:1)欢迎%ai_name%串门回家~
[%{mount_%uid%}]**%ai_name%%>getValue(action_%uid%, 骑着)%%(mount_%uid%)%%(face_%uid%)%进来
[%{mount_%uid%}]**%ai_name%%>getValue(action_%uid%, 骑着)%%(mount_%uid%)%%(face_%uid%)%进入
[%{mount_%uid%}]**%ai_name%%>getValue(action_%uid%, 骑着)%%(mount_%uid%)%%(face_%uid%)%进入直播
间
[%{mount_%uid%}]**%ai_name%%>getValue(action_%uid%, 骑着)%%(mount_%uid%)%%(face_%uid%)%光临
[%{mount_%uid%}]**%ai_name%%>getValue(action_%uid%, 骑着)%%(mount_%uid%)%%(face_%uid%)%来临
["%(face_%uid%)%!=","%(mount_%uid%)%="]**%ai_name%%(face_%uid%)%进来
["%(face_%uid%)%!=","%(mount_%uid%)%="]**%ai_name%%(face_%uid%)%进入
```

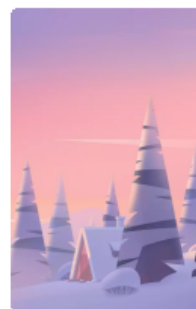
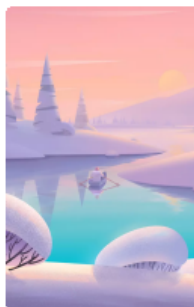


☐ 总开关

黑名单

神奇弹幕·点歌姬

全自动弹幕点歌播放



^点歌[:: ,,]+(.)



☒ 戴勋章点歌



☐ 成功回复



☒ 避免刷歌 1

召唤点歌姬

- ☐ 列表读取接口 最多9行
- ☐ 播放读取接口
- ☐ 歌词读取接口 最多2行
- ☐ 自动暂停其他播放器

☐ 启用

发送

下播啦

[%admin%, %guard%]本场直播结束啦 喜欢的小伙伴点点关注哦! \n我们的直播通知群是: xxxxxxxx\n晚上不见不散(´・ω・`)

☒ 启用

发送

别这么热情\$

>ignoreWelcome(%uid%)\n> 已关闭您的自动欢迎

☒ 启用

发送

^请?叫我\s*(.*)\$

[%(integral_%uid%)%<10000]**>您的积分不足10000, 请多多签到
[%>strlen(%\$1%)% > 5]*>昵称过长, 不能超过5字
[%living%+1]>setNickname(%uid%, %\$1%)\n>设置您的专属昵称为: %\$1%\n>setValue(integral_%uid%, %
(integral_%uid%)%-10000)

☒ 启用

发送

^叫\s*(\S+)\s*(.*)\$

[%admin%;%guard%;%uid%=%up_uid%]*>setNickname(%(%\$1%)%, %\$2%)\n>修改专属昵称成功
>只有房管才能修改别人的专属昵称

☒ 启用

发送

^开启AI回复\$

>execRemoteCommand(开启弹幕回复,0)\n>setValue(reply, 1)\n已开启AI弹幕回复, 发送 "关闭" 结束

☐ 启用

发送



神奇弹幕 v4.0.0

哔哩哔哩直播弹幕姬+答谢姬+点歌姬+录播姬+回复姬



程序简介

神奇弹幕支持弹幕聊天、观众互动、自动点歌、数据统计、原画录播等。

最大的特点是可编程控制，自由制定规则，拥有无限可能！

点此查看[主播案例](#)



开发团队

本程序由 [心义](#) 随手开发，仅供个人学习、研究之用。

内部研讨群: [1038738410](#)

若觉得好用，请[开发者](#) [恰顿饭](#)吧



新闻动态

暂无公告，可前往[官网](#)查看历史动态

附加组件

[视频截图管理](#) [开启大乱斗](#)

[黑听舰长捕捉](#) [跑骚抓人检测](#)

[直播状态查询](#) [歌词字幕投稿](#)

[初]羽Yoko 神奇弹幕

懒一夕智能科技 准备偷塔

[偷塔] 0:1, 赠送2个吃瓜

[己方偷塔] + 2

[对方偷塔] + 10

[反偷塔] 2:11, 赠送10个吃瓜

[对方偷塔] + 10

[己方偷塔] + 10

[己方偷塔] + 10

[反偷塔] 12:21, 赠送10个吃瓜

[己方偷塔] + 10

[对方偷塔] + 10

[反偷塔] 22:31, 赠送10个吃瓜

[对方偷塔] + 10

[反偷塔] 32:41, 赠送10个吃瓜

[己方偷塔] + 10

大乱斗 胜利: 42 vs 41

懒一夕智能科技 赠送 吃瓜 × 42

¥ 4.2 元

用户主页: LV12

粉丝勋章

消息记录: 166条

关注:27,粉丝:7043

播放:10331,点赞:3273

禁言1小时

禁言720小时

永久禁言

添加特别关心

添加强提醒

设置专属昵称

设置用户备注

不自动欢迎

不自动回复

文字

设置

删除

隐藏

皎然记

小义定制播放器

司夏

Q

| | 标题 | 艺术家 | 专辑 | 时长 |
|---------------------------------|----------------------|-------------|--------------------|-------|
| 皎然记 - 司夏 | 杏花弦外雨 | CRITTY/司夏 | 弦上春雪 | 03:57 |
| 千秋此意 - 司夏/汐音社 | 千秋此意 | 司夏/汐音社 | 人间词话 | 03:45 |
| 杏花弦外雨 - CRITTY/司夏 | 吹灭小山河 (翻自 国风堂) | 司夏 | 吹灭小山河 | 03:05 |
| 为龙 (翻自 河图) - 司夏 | 皎然记 | 司夏 | 原创 长夏闻浅歌 | 03:44 |
| 倾尽天下 - 司夏 | 与你最后的夏天 | 司夏 | 与你最后的夏天 | 05:52 |
| 缘生意转 - 司夏/河图 | 芊芊 | 司夏 | 热门华语280 | 03:58 |
| 姑苏一梦 (天下IP×苏州博物馆联动曲) - 化鹤归 - 司夏 | 卷珠帘 | 司夏 | 热门华语262 | 02:46 |
| 归寻 (翻自 等什么君) - 司夏 | 临水照花 | 司夏 | 热门华语270 | 01:51 |
| 少年先锋队队歌 - 司夏 | 少年先锋队队歌 | 司夏 | 热门华语278 | 01:52 |
| 临水照花 - 司夏 | 为龙 (翻自 河图) | 司夏 | 热门华语274 | 04:00 |
| 卷珠帘 - 司夏 | 归寻 (翻自 等什么君) | 司夏 | 归寻 | 03:15 |
| 芊芊 - 司夏 | 化鹤归 | 司夏 | 原创 长夏闻浅歌 | 04:58 |
| 与你最后的夏天 - 司夏 | 姑苏一梦 (天下IP×苏州博物馆...) | 司夏 | 《姑苏一梦》(天下IP×苏州博... | 03:09 |
| 吹灭小山河 (翻自 国风堂) - 司夏 | 缘生意转 | 司夏/河图 | 澈愿记·澈 | 05:11 |
| | 燃尽人间色 | 慕寒/司夏/河图/子弥 | 燃尽人间色 | 04:39 |
| | 多情种 (翻自 胡杨林) | 司夏 | 多情种 | 04:49 |
| | 白衣渡我 | 司夏 | 热门华语232 | 04:36 |
| | 皎然记 | 司夏 | 热门华语280 | 03:44 |

皎然记

司夏

▷ 00:00 ❤️

03:44 🔊

→ 🔊



说明书

使用技巧

绿色版，开箱即用，输入房间号自动连接。

实时弹幕

实时弹幕支持如同聊天软件的水群小窗，**Shift+Alt+D** 快速弹出（非隐藏时有效），ESC 返回之前的程序。

右键菜单中有许多的操作，可自行研究。

若是上班摸鱼，可以设置闲聊模式（仅显示弹幕）+ 简约模式（去头像、彩色），背景全透明，文字半透明灰色，不注意绝对很难看出来。

设置账号Cookie

如果要发送弹幕，必须登录账号。目前有**二维码登录**和**手动设置Cookie**两种方式。

方式一：二维码登录

直接点开菜单中的“账号——二维码登录”，使用哔哩哔哩APP扫码确认。

方式二：使用浏览器Cookie登录

1. 浏览器上按 **F12** 打开【开发者工具】，Network - XHR
2. 进入或刷新 [B站主页](#)（没登录的需登录），Name多了一排，点其中任意一个（若找不到Cookie就换个）
3. 复制如下图的cookie至右上角登录账号的菜单中的“设置Cookie”，即可进行发送。

- 普通的欢迎： `%ai_name%, %greet%%tone/punc%`
将发送：某某某，早上好啊~
- 欢迎舰长： `[%guard%]*欢迎%guard_name% %ai_name% 回家~`
- 一周至一个月没来： `[%come_time%>%timestamp%-3600*24*30, %come_time%<%timestamp%-3600*24*7, 0]*%ai_name%, 太久没来了哦~`

v3.5.7修改：为了保证运行效率，一周没有进入直播间的用户记录将自动清除

- 大乱斗对面来人： `[%pk_opposite%]***欢迎%ai_name%串门哦~`

示例：只欢迎戴自己勋章

未戴粉丝勋章或者不是本直播间粉丝勋章的用户，不自动欢迎；除非是舰长或设置为强提醒：

```
[%guard% = 0, %anchor_room_id!=%room_id%, !%strong_notify%]**
```

两个 * 表示优先级，覆盖没有 * 或1个 * 的弹幕，空弹幕表示不发送。

示例：指定用户欢迎语

指定某一用户的进场通知：

```
[%uid%=12345678]***欢迎%ai_name%, 说好的今天上总督呢~
[%uname%=用户昵称]***欢迎%ai_name%, 今天记得上总督哦~
```

推荐使用UID的方式，不会变更；临时用可用uname。

其中表示优先级的 * 数量务必要比其余弹幕多。

送礼答谢

- 普通的答谢： `谢谢 %ai_name% 的%gift_name%~`
- 小礼物不理： `[%gift_gold%=0, %gift_silver%<1000, %gift_num% < 10]**`
也可以写作： `[%coin_type%=0, %gift_coin%<1000, %gift_num% < 10]**`
1000以下的银瓜子礼物且数量小于10，则忽略。例如忽略6个小心心，但24个小心心则感谢
- 超过80元的礼物： `[%gift_gold%>=80000]*哇噢！感谢 %ai_name% 的%gift_name%! \n老板大气!!!!!!`
\n 表示换行，将分作两条弹幕发送（可以配合 <h1> 放大字体，仅部分弹幕插件有效）
- 跳过机器人送的吃瓜： `[%uid%=my_uid%,%origin_gift_name%=吃瓜]**`

示例：指定礼物专属答谢

赠送小电视飞船，发送指定弹幕：

```
[%gift_name%=小电视飞船]**哇!!! 谢谢%ai_name%带大家上太空~
```

示例：不答谢禁言用户

提升优先级，空弹幕，即屏蔽。

```
[%blocked%]****
```

关注答谢

示例：特别关注

```
[%special%]*(cd21:0)谢谢%ai_name%的特别关注！  
感谢 %ai_name% 的关注%punc%
```

测试时不答谢关注的一个重要原因，是因为刚取消关注就重新关注了，短期内不会答谢！

定时任务

定时发送随机弹幕、执行命令。

示例：今日是否有大航海

```
[%today_guard%=0]今天XX等到新的舰长了吗？\n老板们救救可怜的XX吧~  
[%today_guard>0]今天XX等到了新的舰长，还有老板想上船嘛
```

今天的相关数据都需要在程序连接房间时才有效

示例：定时联网

如果想要定时向某个API发送数据，可通过定时任务结合 `connectNet(url)` 命令

```
>connectNet(要发送数据的网址)
```

示例：定时监测屏幕内容

比较不同时候的屏幕显示内容是否一致。

添加保存的代码，需要手动点击发送：

```
>saveScreenShot(0, 100, 100, 200, 200, D:/test.png)
```

该代码会把矩形（100，100，200*200）这个范围的截图保存到 `D:/test.png` 的位置。

添加定时 1 秒，代码：

```
[%>compareScreenShot(0, 100, 100, 200, 200, D:/test.png)% ==  
100]*>localNotify(相同)  
  
>localNotify(不相同，相似度：%>compareScreenShot(0, 100, 100, 200, 200,  
D:/test.png)%)
```

`%>compareScreenShot(0, 100, 100, 200, 200, D:/test.png)% == 100` 的意思是将当前（100，100，200*200）的截图和之前保存的截图进行比较。

如果显示内容没变，则会打印出“相同”的文字。相似程度从 0 到 100，按像素进行比较，0 为完全不同，100 为完全相同。

有多种算法，并且支持不同参数：

- 按 pixel（默认）：压缩到 8×8 大小后逐一按像素比较
`compareScreenShot(screenId, x, y, w, h, path, pixel)`
返回 0 ~ 100 整数的相似度，越高则越趋于相同。
- 按 pixel 与一定范围内的差值
`compareScreenShot(screenId, x, y, w, h, path, pixel, threshold)` 其中 threshold 为

0 到 255 之间的整数，两张图相同位置的 R/G/B 差值在阈值内的视为同一个点。

返回 0 ~ 100 整数的相似度，越高则越趋于相同。

- AHash

```
compareScreenShot(screenId, x, y, w, h, path, ahash)
```

返回 0 ~ 64 整数的相似度，一般当 ≥ 60 的时候可视为同一张图。

- DHash

```
compareScreenShot(screenId, x, y, w, h, path, dhash)
```

返回 0 ~ 64 整数的相似度，一般当 ≥ 60 的时候可视为同一张图。

- PHash

```
compareScreenShot(screenId, x, y, w, h, path, phash)
```

尚未实现

自动回复

接收到某个关键词，发送对应的回复。

关键词支持正则表达式，使用 `%%x%` 来获取匹配到的文本，其中 `x` 为索引，`%%0%` 为接收的整条弹幕，`%%1%` 为捕获的第一对括号中的文字，`%%2%` 为第二对，下同。

示例：关键词自动禁言

有专门的禁言选项卡，此示例仅仅是展示使用方式

添加需要禁言关键词 `xxx`，支持正则表达式。设置回复：

```
>block(%uid%)
```

示例：房管远程弹幕禁言

默认仅主播与机器人账号支持弹幕禁言，此方法可提升至房管（或其他特定条件用户）：

添加回复：`^((禁言|解禁|解除禁言|取消禁言) .+|撤销禁言)$`，添加动作：

```
[%admin%]>execRemoteCommand(%text%)
```

示例：不要欢迎我

添加表达式：`别这么热情`，添加回复：

```
>ignorewelcome(%uid%)\n>已关闭您的自动欢迎
```

只要弹幕中包含“别这么热情”，则以后都不会自动欢迎

示例：设置自己专属昵称

添加自动回复表达式：`^请?叫我\s+(.*)$`，添加回复：

```
>setLocalName(%uid%, %%1%)\n>设置您的专属昵称为：%%1%
```

当用户发送弹幕：“叫我 小明”或“请叫我 小明”，程序自动设置其专属昵称为“小明”。

可以使用 `%local_name%` 或者 `%ai_name%` 获取。

示例：修改指定用户昵称

添加自动回复表达式：`叫(\s+)\s+(.*)`，添加回复：

```
[%admin%]>setLocalName(%($1%)%, %$2%)\n>修改专属昵称成功
```

浏览器插件

开启“扩展—网络服务”，之后于直播姬中添加浏览器并访问网址，即可打开插件页面，无缝嵌入直播界面。

网址为：`http://localhost:5520` + 路径。

目前支持以下功能：

| 说明 | 路径 |
|------------|-----------------------|
| 点歌列表 | /music/index.html |
| 正在播放的歌曲 | /music/playing.html |
| 正在播放的歌曲的歌词 | /music/lyrics.html |
| 送礼物小动画 | /gift/index.html |
| 电影结束效果答谢 | /gift/end_thanks.html |

例如点歌列表，默认的连接是：

```
http://localhost:5520/music/index.html
```

更多功能可自行定制，接口与开发文档见底部“[Web开发接口](#)”。

远程控制

主播或者机器人账号在直播间内发送如 `关闭欢迎`、`开启录播` 等命令，可远程控制在服务器上的机器人（本地没必要使用远程命令）。

- 关闭/开启功能：主开关
- 关闭/开启欢迎
- 关闭/开启关注答谢
- 关闭/开启送礼答谢
- 关闭/开启禁言：新人关键词自动禁言
- 关闭/开启点歌
- 关闭/开启点歌回复
- 关闭/开启定时任务
- 关闭/开启自动连接
- 关闭/开启弹幕回复：AI回复弹幕
- 关闭/开启录播
- 关闭/开启粉丝档案
- **禁言 [昵称] [小时]**：房管权限，根据[昵称]（允许一部分，根据弹幕记录自动倒找发言人）禁言某用户，[小时]可省略，默认为自动禁言的时长（禁言相关都需要在设置中开启“**启用禁言（主播/房管）**”），例如 `禁言 小明 3`，注意空格分隔
- **解禁 [昵称]**：解除禁言，[昵称]支持用户名中的一部分
- **撤销禁言**：依次解禁最近禁言的用户
- **永久禁言 [昵称]**：永久禁言某用户，可通过“解禁 [昵称]”、“撤销禁言”远程解除
- 关闭机器人：全局关闭（将无法远程唤醒）

弹幕样式

支持自定义实时弹幕的CSS样式，实时弹幕的右键菜单中，设置->标签样式。

圆角矩形

```
background: white;
padding: 5px;
border-radius: 10px;
```

气泡图片

```
padding: 10px;
border-image: url(../bubbles/bubble1)
```

其中图片可使用本地绝对路径，例如：

```
border-image: url(C:/Path/To/Image.png)
```

类型选择

支持按不同类型设置不同的样式，样式如下：

- msg：一些提示
- danmaku：弹幕（样式不影响左边头像显示）
- gift：送礼
- welcome：进入
- order-song：点歌提示文字：[点歌] 歌名
- guard-buy：开通/续费舰长
- welcome-guard：舰长进入
- attention：关注
- block：禁言
- share：分享直播间
- super-chat：醒目留言

以下是一个示例：

弹幕为默认气泡；礼物、上舰为圆角矩形卡片

```
#danmaku {
  border-image: url(../bubbles/bubble1);
  padding: 10px;
}
#gift, #guard-buy {
  background: #FFDAB9;
  padding: 5px;
  border-radius: 10px;
}
```

多开与备份

无限多开

本程序支持无限多开，在程序根目录下创建“green_version”文件，即是绿色版（默认），不同文件夹中的神奇弹幕使用不同的数据。删除文件夹时，程序及其使用数据将全部删除，不会留下痕迹。

若未设置成绿色版，则不同文件夹的神奇弹幕使用同一套数据，多开可能会有冲突。删除文件夹时，使用数据不会删除。此时升级很方便，新下载的安装包删除“green_version”即可恢复之前数据。

自动备份、还原

每次启动的时候会将程序配置（settings.ini）和使用数值（heaps.ini）各加上当前时间备份至“backup/”文件夹，若不小心被其他版本覆盖，可将备份的这两个文件去掉文件名中的时间，放到程序根目录下。

调试日志

如果经常遇见问题，请按以下方法开启日志：

在程序根目录的 `settings.ini` 文件中，`[runtime]` 分组下（没有则自己新建），添加 `debugToFile=true`。

重启神奇弹幕，根目录下会出现 `debug.log`，这就是调试日志。

盘查问题时需要将这个文件交给开发者。

可编程设计

通过一系列用户自由设计的代码规则，将工作分解成定义良好的任务，制定其触发条件和执行顺序，按照一定的规则和过程来执行这些任务，这就是 workflow。

workflow 的意义

直播间实施 workflow 管理所带来的好处是非常明显的，这包括提高直播间运营效率、改善人气资源利用、提高直播运作的灵活性和适应性、提高工作效率、集中精力处理弹幕聊天、跟踪观众互动过程、量化粉丝粘度、统计礼物数量，充分发挥现有直播数据的作用。实施 workflow 将达到缩短互动等待周期、改善弹幕流程、优化并合理利用资源、减少人为差错和延误、提高吸粉效率等目的。

总结可编程 workflow 带来的好处，可以归纳为以下几点：

1. 要处理的事项自动回复
2. 不需要对观众的数据进行手动整理
3. 主播只需要将精力集中在关心的互动上
4. 保存所有历史记录
5. 随时看到统计数据
6. 达到无纸化办公的目标
7. 完全支持分布式架构，多设备同时工作
8. 科学管理更近一层，直播效率明显提高
9. 直播间的核心竞争力将有所提升
10. 通过流程自动化与数据库集成，以及各类表单统计查询功能，提高决策能力

workflow 的实现需要三个基本步骤：映射、建模和管理。映射是第一个步骤，其首要任务是确定直播间内全部现有的手工和自动化的业务流程；建模则是开发一个有助于与观众交互的模型。第三阶段是软件实施以及跨越多项业务逻辑的无缝系统集成。

为了确保 workflow 系统能够无缝地衔接到直播间中，用户必须遵从已经定义好的、经过实践确认的行之有效的 **脚本规则**，并且在每个脚本都必须有可以度量的结果。在实际使用中，根据主播的情况来制定符合要求的脚本，在易用性和功能需求之间达到平衡。

简单介绍

弹幕的候选列表中，支持一系列的可编程变量、简单逻辑运算、简单算术运算。

使用两个 % 包括的英文，则为变量。数据变量例如 %uname%，与当前的一系列数据相关，例如礼物价值、今日人数等等；招呼变量例如 %greet%，**自动替换**为当前时间段对应的招呼语，例如候选项表达式 %ai_name%，%greet%%tone/punc%，在早上可能是“早上好啊，某某某”，在下午可能是“某某某，下午好~”，在晚上又会是“晚饭吃了吗，某某某~”。

示例：

- 简单的欢迎： 欢迎 %ai_name% 光临~
- 动态语气词： %ai_name%，%greet%%tone/punc%

看下去，有更多例子。

如无特殊说明，下述代码的符号均以**英文符号**为准，如 []、()、,，不支持中文的 【】、（）、，等。

自定义变量

在“扩展”页面，右上角的菜单中选择“自定义变量”，在打开的对话框中设置自己需要的变量。

每一行使用“变量名 = 变量值”，所有代码块中的变量名都会替换为变量值。

一般来说，默认包含以下变量：

- %upname%：主播昵称

这些自带变量可能会在自带的代码中出现，请务必修改。

数据变量

同上，都需要带上百分号，例如 %var%，表格中皆省略。

目前已支持中文，可直接使用形如 %用户ID% 的格式。

| 变量 | 中文 | 说明 |
|-------------|--------|----------------------------|
| uid | 用户ID | 是一串数字，确定唯一用户 |
| uname | 用户昵称 | 需要用户值，例如定时消息，则只是空字符串 |
| username | 用户昵称 | 和上面一模一样 |
| nickname | 用户昵称 | 同上 |
| ai_name | 用户智能昵称 | 优先专属昵称，其次简写昵称，无简写则用原昵称 |
| local_name | 用户专属昵称 | 实时弹幕中右键-设置专属昵称 |
| simple_name | 用户简写昵称 | 去除前缀后缀各种字符 |
| room_id | 直播间ID | |
| room_name | 直播间标题 | |
| up_uid | 主播ID | |
| up_uname | 主播名字 | |
| my_uid | 机器人ID | |
| my_uname | 机器人名字 | |
| csrf | csrf | 用户的csrf token |
| level | 用户等级 | 进入直播间没有level |
| text | 弹幕消息 | 若有换行，则替换为 <code>%n%</code> |

| 变量 | 中文 | 说明 |
|------------------|-----------------|---|
| url_text | URL编码的弹幕消息 | 针对弹幕消息进行URL编码，用于网址传参，避免一些特殊文本 |
| come_count | 用户进入次数 | 长时间没来会被清空 |
| come_time | 用户上次进入时间 | 10位时间戳，第一次进来是0；自动清理一周没来的。如果是串门回来，则是刚跑到对面串门的时间 |
| come_time_delta | 进入时间差 | 当前时间 - 用户上次进入时间 |
| gift_id | 礼物ID | 盲盒是爆出来的礼物 |
| gift_gold | 礼物金瓜子 | 非送礼答谢则没有，盲盒是原始价格 |
| gift_silver | 礼物银瓜子 | |
| gift_coin | 礼物瓜子 | 不分金瓜子银瓜子 |
| coin_gold | 是金瓜子礼物 | 金瓜子是1，银瓜子是0 |
| gift_name | 礼物名字 | 如果设置了别名，则使用别名 |
| origin_gift_name | 礼物原始名字 | 未设置别名时等同于gift_name |
| gift_num | 礼物数量 | |
| gift_multi_num | 带单位的礼物数量 | 如果为1个，则忽略，为空文本 |
| guard_buy | 开通大航海 | 上船消息，gift_name：舰长/提督/总督；guard：3 舰长/2提督/1总督 |
| guard_frist | 初次上船 | 初次1，重新上船2，其余0 |
| total_gold | 用户总共金瓜子 | 该用户一直以来赠送的所有金瓜子数量 |
| total_silver | 用户总共银瓜子 | 同上 |
| anchor_room_id | 粉丝勋章房间ID | 进入、弹幕才有粉丝牌，送礼只能获取粉丝牌名字 |
| medal_name | 粉丝勋章名称 | 同上 |
| medal_level | 粉丝勋章等级 | 同上 |
| medal_up | 粉丝勋章主播昵称 | 只有弹幕消息有 |
| nickname_len | 用户昵称长度 | |
| giftname_len | 礼物名字长度 | |
| name_sum_len | 用户昵称长度+礼物名字长度 | |
| ainame_sum_len | 用户智能昵称长度+礼物名字长度 | |

| 变量 | 中文 | 说明 |
|------------------|---------|------------------------------------|
| danmu_longest | 弹幕字数上限 | 当前账号的最长弹幕 |
| new_attention | 新关注 | 最近50个关注内 |
| guard_buy_count | 上船次数 | 舰长+1、提督+10、总督+100（只统计程序运行时），0为初次上船 |
| pkimg | 大乱斗中 | 是：1，否：0 |
| pk_video | 视频大乱斗 | 是：1，否：0 |
| pk_id | 大乱斗编号 | |
| pk_room_id | 对面房间号 | 未在PK中则为空，下同 |
| pk_uid | 对面主播ID | |
| pk_uname | 对面主播昵称 | |
| pk_opposite | 对面进入 | 需开启串门提示 |
| pk_view_return | 去对面串门回来 | 需开启串门提示 |
| pk_count | 匹配次数 | 未在PK中为0，下同 |
| pk_my_votes | 己方积分 | |
| pk_match_votes | 对方积分 | |
| area_id | 分区ID | |
| area_name | 分区名字 | |
| parent_area_id | 主分区ID | |
| parent_area_name | 主分区名字 | |
| today_come | 今日进入人次 | 每个人可能重复进入 |
| today_newbie_msg | 今日新人人数 | |
| today_danmaku | 今日弹幕总数 | |
| today_fans | 今日新增粉丝数 | |
| today_gold | 今日金瓜子总数 | |
| today_silver | 今日银瓜子总数 | |
| today_guard | 今日上船人次 | 续几个月算多次 |
| today_max ppl | 今日最高人气 | |
| popularity | 当前人气 | |
| guard_count | 大航海人数 | |
| fans_count | 总粉丝数 | |
| fans_club | 总粉丝团 | |

| 变量 | 中文 | 说明 |
|-----------------|----------|-----------------------------------|
| admin | 房管 | 只有弹幕、进入才有 |
| guard | 大航海级别 | 只有弹幕消息/购买舰长/舰长进入有；普通0，舰长3，提督2，总督1 |
| guard_name | 大航海身份 | “舰长”/“提督”/“总督”，非船员为空 |
| admin_or_guard | 是否是房管或舰长 | 只有弹幕消息有，0或1 |
| online_rank | 高能榜 | 返回高能榜排名，若不在高能榜上，则为0 |
| vip | 姥爷 | 同上 |
| svip | 年费姥爷 | 同上 |
| uidentity | 正式会员 | 同上 |
| iphone | 手机号认证 | 同上 |
| number | 数量 | 各种数量的通用写法，如礼物数量等 |
| time_hour | 当前小时 | |
| time_minute | 当前分钟 | |
| time_second | 当前秒 | |
| time_day | 当前日期 | |
| time_month | 当前月份 | |
| time_year | 当前年份 | |
| time_day_week | 当前星期 | 1~7 |
| time_day_year | 今年第几天 | |
| timestamp | 当前时间戳 | 10位时间戳，可用于比较进入时间、多久没来等 |
| timestamp13 | 当前时间戳13 | 13位时间戳 |
| in_game_users | 在游戏用户中 | uid在gameUsers[0]中（高性能，但重启清空） |
| in_game_numbers | 在游戏数值中 | uid在gameNumbers[0]中 |
| in_game_texts | 在游戏文本中 | text在gameTexts[0]中，空格等都需要一模一样 |
| app_name | 程序名字 | 返回exe名字（不带后缀），可用于多开区分 |
| app_path | 程序目录 | 数据路径，绿色版为安装目录 |
| www_path | 网站目录 | 数据路径下的www目录的绝对路径 |
| server_domain | 网站域名 | 默认为 <code>http://localhost</code> |
| server_port | 网站端口 | 默认为 5520 |
| server_url | 网站网址 | 等于server_domain:server_port |

| 变量 | 中文 | 说明 |
|------------------|----------|--|
| living | 直播中 | 直播：1，轮播：2，未开播：0 |
| is_up | 是主播 | |
| is_me | 是机器人 | |
| is_room_medal | 戴本勋章 | 戴本直播间的粉丝勋章 |
| care | 特别关心 | 是：1，否：0 |
| strong_notify | 强提醒 | 是：1，否：0 |
| not_welcome | 不自动欢迎 | 不欢迎：1，欢迎：0 |
| not_reply | 不自动回复 | 是否是机器人发的弹幕，是：1，否：0 |
| blocked | 被禁言 | 禁言：1，未禁言0 |
| playing_song | 当前歌曲 | 点歌姬没有播放歌曲则为空 |
| song_order_uname | 当前歌曲点歌者 | 同上 |
| random100 | 随机100 | 1~100的随机数，同代码块中的都将随机替换为同一个数，可用于设置抽奖概率 |
| special | 特别关注 | 关注答谢里可用，判断特别关注 |
| spread | 推广方式 | 一般为空，可能是“星光推广” |
| pk_magical_room | 对面也用神奇弹幕 | 是：1，否：0 |
| cd0~cd99 | | 冷却通道，等同于 <code>%>cd(X)%</code> |
| wait0~wait99 | | 等待通道，等同于 <code>%>wait(X)%</code> |
| local_mode | 本地模式 | 是：1，否：0 |
| repeat_10 | 重复弹幕10 | 最后一条与最近10条弹幕是否重复，是：1，否：0 |
| playing_tts | 播放TTS | 是否正在朗读弹幕/文字，是：1，否：0 |
| mouse_x | 鼠标X | 获取当前鼠标的全局X坐标，相对于主屏幕左上角，若在左边副屏则 x 可能是负数 |
| mouse_y | 鼠标Y | 同上 |

特殊用法

打破“仅直播时发送”

仅在 **定时任务**、**自动回复**、**事件动作** 中有效，在开启“**仅直播时回复**”开关后，下播后所有自动弹幕都将不发送。

此时在需要发送的弹幕框中，其 `任务/回复/动作` 中带有 `%living%` 选项，则在非直播时也会激活。

例如自动回复的动作为：`[%living%+1]弹幕文字`，该回复不受直播状态影响。

如果开启了“定时连接B站服务器”，那么超出时间段下播后神奇弹幕不会接收所有弹幕，所有功能暂停。

自定义bilibili接口

`%csrf%`为用户cookie，用于连接B站的API，结合本程序的 `connectNet` (GET)、`postData` (POST)，向服务器发送指定数据。

招呼变量

| 变量 | 描述 | 示例 |
|-----------|---------|---------------------------|
| hour | 时辰 | 早上/中午/下午/晚上 |
| greet | 招呼 | 您好、早上好、晚上好 |
| all_greet | 带语气词的招呼 | 你好、早上好啊、午饭吃了吗、晚上好呀、怎么还没睡~ |
| tone | 语气词 | “啊”或“呀” |
| lela | 语气词 | “了”或“啦” |
| punc | 标点 | “~”或“! ” |
| tone/punc | 语气词或标点 | 上面两项，适用于和%greet%结合 |

变量计算

| 变量 | 描述 | 注意事项 |
|----------------------------|----------------|-----------------------------|
| <code>%{key}%</code> | 获取配置文件中的值 | key为setValue(key)中的键值，未设置为空 |
| <code>%[exp]%</code> | 简单的数值计算 | 支持加减乘除、括号，不支持小数 |
| <code>%<exp>%</code> | JavaScript脚本计算 | 允许声明变量与各种计算，断句换行使用； |
| <code>%(name)%</code> | 将用户昵称转换为uid | 需要实时弹幕上显示才有效，允许部分昵称 |

`%{key}%` 如果是一个未设置的值，那么将会是空字符串，如果要转化为数字 `0`，可以使用 `%[%{key}%+0]%` 的方式。

`%{key}%` 支持默认值，格式为：`%{key|def}%`，如果key不存在，则会使用def的值，可以是数字或者任意字符串，例如 `%{key|0}%` 能实现上面的效果。

示例：礼物价值

谢谢%ai_name%赠送了价值%[%gift_gold%/1000]%元的%gift_name%!

常量

| 常量 | 描述 |
|------------------|--|
| <code>%n%</code> | 替换为换行符 \n，目前仅支持： <code>postData()</code> 、 <code>postJson()</code> 、 <code>writeTextFile()</code> 、 <code>appendFileLine()</code> 、 <code>sendToSockets()</code> 、 <code>sendToLastSocket()</code> 、 <code>runCommandLine()</code> 、 <code>sendLongText()</code> |

代码展开

目前只有一个：`#traverseJson(json_path, code)`，遍历JSON的指定内容，并展开为代码列表。

用来遍历不确定的 JSON 数据，可以是 JSON 对象，或者 JSON 数组。

参数说明：

- path: 本程序语法中的 JSON 路径，不带 %，如 `.data?.array?`，对应具体的对象或数组
- code: 要执行的代码，需要进行转义：% 写作 `\%`，`\n` 写作 `%n%` 等

如果是 JSON 数组：

- `code` 中引用到的 JSON 数组索引使用 `\%i%` 来获取（实际上 `%i%` 也可以）

如果是 JSON 对象：

- `code` 中引用到的 JSON 键值对使用 `\%key%` 来获取

示例：知识库列表

使用 AI 检查不定数量的知识卡片，其JSON 格式：

```
{
  "type": "回答",
  "msg": "测试内容",
  "tip": "这是测试内容",
  "knowledge": [
    {"name": "知识1", "content": "这是知识1的内容"},
    {"name": "知识2", "content": "这是知识2的内容"},
    {"name": "知识3", "content": "这是知识3的内容"}
  ]
}
```

如果需要展开显示所有 `knowledge` 中的内容：

```
#traverseJson(.knowledge, >localNotify(【\%.knowledge.\%i\%.name\%】
\%.knowledge.\%i\%.content\%))
```

会展开为代码：

```
>localNotify(【%.knowledge.0.name%】%.knowledge.0.content%)\n\
>localNotify(【%.knowledge.1.name%】%.knowledge.1.content%)\n\
>localNotify(【%.knowledge.2.name%】%.knowledge.2.content%)
```

最后运行结果：

```
【知识1】这是知识1的内容
【知识2】这是知识2的内容
【知识3】这是知识3的内容
```

函数计算

按指定格式，获取动态的数值，格式：`%>func(args)%`

| 函数 | 中文 | 描述 |
|----------------------------|-------|--|
| cd(channel) | 冷却通道 | 获取冷却通道剩下的秒数 |
| wait(channel) | 等待通道 | 获取等待通道不是自己的弹幕数量 |
| time(format) | 格式化时间 | 当前时间转换为数值，如 yyyy-MM-dd |
| unameToUid(uname) | 查找用户名 | 由部分昵称倒找弹幕记录，获得UID |
| ignoreEmpty(text) | 忽略空 | 当text为空字符串或者各种括号、引号等包裹的空白内容时进行忽略，若非空则返回原text |
| inputText(title, default) | 输入文本 | 输入文本，两个参数都可省略 |
| strlen(text) | 取文本长度 | 一串文字的长度 |
| trim(text) | 删首尾空 | 去掉字符串首尾的空格和制表符 |
| substr(text, left, length) | 取子串 | 获取文字的一部分 |

| 函数 | 中文 | 描述 |
|----------------------------|--------|--|
| simpleName(name) | 昵称简化 | |
| simpleNum(number) | 数值简化 | |
| inGameUsers(listId, uid) | 在游戏用户中 | listId可省略。程序重启数据会清空 |
| inGameNumbers(listId, num) | 在游戏数值中 | listId可省略，程序重启数据仍在 |
| inGameTexts(listId, text) | 在游戏文本中 | listId可省略，程序重启数据仍在 |
| getValue(key, def) | 取变量值 | 等同于 <code>%{key}%</code> ，默认值def可省略 |
| random(min, max) | 取随机数 | 包含两端数字。一个参数时默认为0~max。允许超出2个参数，为了区分多个相同范围的不同随机数结果 |
| randomArray(a, b, c, d...) | 随机数组 | 任意多参数，随机返回其中一个 |
| filterReject(filter) | 过滤器拒绝 | 被对应filter拒绝则返回1,否则返回0（参考过滤器示例） |

| 函数 | 中文 | 描述 |
|--------------------------------|--------|--------------------------|
| inFilterList(filter, content) | 在过滤列表中 | 包含在空格分隔的词库中则返回1（参考过滤器示例） |
| inFilterMatch(filter, content) | 在过滤正则中 | 满足正则则返回1（参考过滤器示例） |
| fileExists | 文件存在 | 有这个文件则返回1，否则0 |
| abs(val) | 取绝对值 | |
| log2(val) | 取对数2 | |
| log10(val) | 取对数10 | |
| pow2(val) | 取平方 | |
| pow(val, a) | 取乘方 | |
| pasteText() | 粘贴文字 | 获取剪贴板的文本 |

| 函数 | 中文 | 描述 |
|------------------------------------|----------|---|
| getScreenPositionColor(wid, x, y) | 获取屏幕位置颜色 | wid为屏幕ID（一般为0），x y 为横纵坐标，返回例如 #f0f0f0 的颜色格式 |
| getWindowPositionColor(name, x, y) | 获取窗口位置颜色 | name 为窗口名字（可以是部分）或者句柄ID，不能是最小化窗口 |
| execReplyResult(text) | 执行回复结果 | 获取满足text的第一个回复的执行结果；结果中若存在命令则会执行，若有换行符“\n”则会返回用它分隔的单行文字 |
| execEventResult(event) | 执行事件结果 | 获取指定事件的执行结果，重复则取第一个，同上 |
| readTextFile(fileName) | 读取文本文件 | 读取文本文件中的所有内容，所有换行符 \n 将会被替换为 %n% |
| getTextFileLine(fileName, line) | 获取文本文件行 | 读取文本文件中的第 line 行，行数从 1 开始 |

| 函数 | 中文 | 描述 |
|---|----------|---|
| getTextFileLineCount(fileName) | 获取文本文件行数 | 读取文本文件中的所有行数（以 \n 为准） |
| compareScreenShot(screenId, x, y, w, h, path[, other args]) | 比较窗口截图 | 比较当前窗口的截图与指定图片文件的相似度。 screenId:屏幕ID，只有一个屏幕则为0；x/y/w/h:坐标与宽高；path:要比较的文件路径。 返回一个0到100的数字，表示相似程度。具体请参考 示例 |
| getScreenWidth(screenId) | 获取屏幕宽度 | 获取指定显示器的屏幕宽度。如果不指定屏幕ID，则默认为0 |
| getScreenHeight(screenId) | 获取屏幕高度 | 同上 |
| findWindow(窗口标题) | 查找窗口 | 根据完整的窗口标题或类名来查找顶级窗口（不搜索子窗口），返回一个窗口句柄（hwnd），可用于作为修改窗口的命令的参数。结合 spy++ 工具来获取标题或类名 |
| getForegroundWindow() | 获取前景窗口 | 获取当前具有焦点的窗口的句柄（hwnd） |
| isWindowFullScreen(hwnd) | 窗口是否全屏 | 判断窗口是否全屏（不包含最大化），可用于判断游戏全屏、视频全屏 |

| 函数 | 中文 | 描述 |
|------------------------------|----------|---------------------------|
| getCursorPos(x/y) | 获取鼠标位置 | 获取当前鼠标的位置，参数为 x 或 y 字符串 |
| getWindowRect(hwnd, x/y/w/h) | 获取窗口位置 | 获取指定窗口几何坐标的值 |
| getWindowFromPoint(x, y) | 获取坐标所在窗口 | 获取屏幕上坐标为(x,y)的位置指向的最前面的窗口 |

以获取时间为例：

```
当前时间： %>time(yy-MM-dd hh:mm)%
```

####

四则运算

数字与数字、字符串与字符串之间可进行比较。其中运算符支持 加+、减-、乘*、除/ (向下取整)、取模%、包含~，比较支持 大于>、小于<、等于=、不等于!=、大于等于>=、小于等于<=。

tips:

- 开发人员友好，= 可写作 ==
- 不等于 != 也可以是 <>
- 字符串两端可不用加双引号 "

比较的两边，当都是数字或算数表达式时，自动进行简单的计算（暂不支持复杂格式）。

另外，仅支持在开头的 [] 判断中进行四则运算，在弹幕内容中计算需要使用 %[公式]%

数值修改延迟

该问题在通常情况下可使用“语法糖”来解决。

所有变量的值都是在**触发的一瞬间赋值**的，如果在同一行中前面修改了某一值，那么后面的值也不会跟着修改。请看以下示例：

```
>setValue(count, %[#{count}%+1]%) \n数量: #{count}%
```

将数量count的大小加1并发送数量的弹幕，但显示的count是修改之前的大小。

应使用以下方式，计算两次：

```
>setvalue(count, %[%{count}%+1]%) \n数量: %[%{count}%+1]%
```

另外，由于弹幕冷却时间，所有命令也不是瞬间执行完毕，会按照一定的时间间隔依次执行，可能会导致读取的时候前面命令还没写完。典型的例子就是多个人同时打卡，总次数只会增加一次（但不影响两个人的打卡天数、积分等）。

语法糖

V4.6.0 版本开始，新增一项开关：语法糖，默认开启。

如果要修改一个值，以前的版本是比较复杂的，例如将打开天数+1：

```
>setvalue(data, %[%{daka}%+1]%)
```

而通过语法糖，则只需要简单的相加即可：

```
{daka}++
```

目前开启此项功能后，可支持以下语法糖：

- 赋值：`{var} = 20`，等同于 `>setvalue(var, 20)`
- 加/减/乘/除/取模：`{var} += 10`，等同于 `>setvalue(var, %[%{var}% + 10]%)`
- 自增/自减：`{var}++`、`{var}--`，前者等同于 `>setvalue(var, %[%{var}% + 1]%)`

V5.6.0 版本开始，支持更简单的语法糖：四则运算不需要再使用等号。

例如：`{var} += 10` 可以直接简写成为 `{var} + 10`，直观明了。

字符串运算

在语法糖模式下，字符串也支持简单的运算。

- `A+B`：两个字符串A和B相加
- `A-B`：字符串A中去除一次子串B
- `A/B`：字符串A中去除所有子串B
- `A/n`：n为次数，字符串A的长度缩减到1/n，向下取整，例如“哈哈/3”变为“哈”，长度不足时为“空字符串”
- `A*n`：n为次数，字符串A重复n遍

条件判断

与编程语言相似的算法，每一行使用 `[]` 开头，则方括号中的内容会识别为 条件表达式，使用 `,` 或 `&&` 来执行“与”逻辑，使用 `;` 或 `||` 来执行“或”逻辑；“或”的优先级更高。`[]` 中使用 `%val%` 作为变量值，例如 `[%level%>10]` 弹幕内容，则只有当用户等级超过10级时才会被发送 弹幕内容。

(V4.0.6新增) 当条件表达式包含字符串或者正则表达式时，可能会出现符号冲突问题，使用双方括号 `[[条件]]` 或者三方括号 `[[[条件]]]` 来替换单方括号。

示例：

- 用户等级为0级：`[%level%=0]` 或 `[%level% == 0]`
- 粉丝牌等级介于10级到19级之间：`[%medal_level%>=10, %medal_level%<20]`

- 名字为某某某：`[%uname%=某某某]` 或 `["%nickname%"=="某某某"]`
- 现在是黑夜：`[%time_hour% > 17 || %time_hour% <= 6]`
- 带粉丝牌的0级号，或非0级号：`[%anchor_room_id%, %level%]`
- 付费两万（2千万金瓜子）的老板：`[%total_gold% >= 20000000]`
- 一小时内重新进入直播间：`[%come_time% > %timestamp%-3600]`
- 几周没来：`[%come_time%>%timestamp%-3600*24*30 && %come_time%<%timestamp%-3600*24*7]`
- 某个片哥名字匹配：`[[%text% ~ "[\u4e00-\u9fa5]+[\w]{3}[\u4e00-\u9fa5]+"]]>block(%uid%)`

优先级

文本框中内容支持多行，一行为一条候选项，随机发送一条。其中每一条都可以用星号 `*` 开头（若有条件表达式 `[exp]`，则 `*` 位于表达式后面），星号数量越多，则优先级越高。高优先级候选应当带有条件，当满足条件时，发送该条弹幕，并无视掉所有更低优先级的候选。

带有优先级的候选项，会被更高优先级（更多星号）的候选项所覆盖。

例如：

```
感谢 %ai_name% 的%gift_name%，么么哒~
[%gift_gold%>=10000]感谢 %ai_name% 的%gift_name%，老板大气！
[%gift_gold%>=50000]*(cd15:3)哇塞！感谢 %ai_name% 的%gift_name%！\n老板大气！
[%gift_gold%>=150000]**(cd16:1)哇噢！感谢 %ai_name% 的%gift_name%！\n老板大气！！！！！！
```

注意星号，1、2无优先级，即使满足第2项的条件，也是随机发送前两项之一。3优先级超过1、2，因此当满足3的条件并且不满足4时，会发送第3项。而当4的条件满足时，其余更低优先级的候选项都被无视，只发送4。

另外，3、4加入了 `冷却通道`，表示不使用默认的送礼冷却。以4为例，改为冷却1秒，相当于强制发送（B站限制，最快1秒）。

- 当本次礼物的金瓜子总价少于1万时，发送第一项；
- 当本次礼物的金瓜子总价小于5万时，随机发送第一项、第二项；
- 当本次礼物金瓜子总价介于5万到15万时，发送第三项；
- 当本次礼物金瓜子总价超过15万时，发送第四项。

高优先级被忽略情况

高优先级只是说明“**在当前情景下更适合**”，并不代表一定会发送这一条。

当这一条弹幕的字数超过了设置的“弹幕最长长度”，那么高优先级的弹幕会被忽略，转而发送低优先级的。

至于冷却通道，则不会影响优先级。当最高优先级的弹幕尚在冷却中时，不会发送弹幕。

注释

使用 `//` 作为注释符号，直至后面**行尾**或者**弹幕结束标记** `\n`，在发送弹幕时将被忽略。

使用 `///` 作为显眼的注释说明，将会是高亮的颜色，而不是灰色。

也支持多条连续弹幕插入注释，如下：

```
// 单行注释，不会被发送
这是弹幕 // 弹幕后面的注释
第一条弹幕//中间的注释1\n第二条弹幕//中间的注释2\n第三条弹幕
```

软换行

在行尾使用 `\` 可拼接连续的两行，不影响发送内容，示例如下：

```
计算结果： %[1      \
              +2    \
              +3]%
```

多条弹幕

使用 `\n` 来分割过长弹幕，则会分多条弹幕发送，每次延时约1.5s。通过此方式支持**执行多条命令**。

这是两个普通字符，并不是换行符。对于编程人员来说，相当于代码中两个反斜杠加字母n。

受于B站后台的限制，多条弹幕将调整为每隔1.5秒发送一次，数量无上限。

触发多段代码

一个条件（定时、关键词、事件等）可以触发多段代码，在一段代码中也可以使用 `---` 拆分为多段：

```
代码块1
---
代码块2
---
代码块3
```

发送选项

在发送的弹幕前添加英文小括号，里面可包含一些设置项 (option)，设置弹幕发送这一瞬间的属性。

- 冷却通道 (cd)：控制弹幕发送频率
- 等待通道 (wait)：控制自己不连续发送弹幕
- 强制房管权限 (admin)：向普通观众的弹幕开放部分权限
- 子账号 (account)：选择要用作发送弹幕的子账号

举个例子：

```
[某条件]*** (cd20:5,wait12:5)欢迎xxx~
```

发送选项必须在条件、优先级后面，顺序不能变动。多个发送选项使用英文逗号隔开。

与条件的区别

这里以冷却通道为例：

- 条件的方式：`[%cd20%>5]` 某弹幕
- 选项的方式：`(cd20:5)` 某弹幕

同样是判断冷却通道20号的时间有没有满5秒，当满5秒的时候，两者的行为是一致的，都将发送弹幕。

若没到5秒，两者也都不会发送弹幕，但是具体影响上会有一些区别：

- 条件：会判断其他行的弹幕会不会发送，如果可以，则发送出去；
- 选项：不会判断其他弹幕，直接不发送。

条件用来判断发送哪些弹幕，选项用来控制弹幕发送频率。

强制房管权限

选项中添加 `admin` 参数即可，如：

```
[%medal_level% >= 30](admin)>triggerReply(某回复)
```

该选项是用来提升普通用户的权限，即使不是房管，也可以使用一些相同的权限。

此处设置的权限仅用于代码块中的 `%admin%` 判断，不影响具体房管身份以及其他权限。

此处设置的房管权限可通过 `>triggerReply()` 和 `>triggerEvent()` 传递到其他代码。

当然，也可以作为手动点“发送”按钮的一项调整。

冷却通道

v2.9.0 版本新增

为实现不同条件使用不同的冷却时间，新增了“冷却通道”的规则——连续发送弹幕（例如自动欢迎）会有一定冷却时间，但有时候会想要强制发送（例如强制欢迎舰长），或者自定义部分冷却时间（例如在有人连刷小心心的时候，突然有人上船，应当想要立刻通知）。

格式：`cd通道:冷却秒数`

在自定义弹幕的开头使用形如 `(cd25:10)` 的格式，表示 现在如果在上次发送的冷却通道25号的弹幕10秒之后，则发送当前弹幕，即满足其10秒的冷却时间，并不受其他自动弹幕的影响。

示例：强制欢迎

每30秒欢迎一次用户；而若有25级及以上粉丝牌的用户进来，立刻欢迎，除非连续两个25级及以上的用户在5秒内进来，则只欢迎前一人；若是舰长，同上，多条冷却通道之间互不影响。而25级舰长，根据舰长优先级 `**` 超过25级优先级 `*`，会优先发送欢迎舰长的弹幕。

```
(cd10:30)欢迎%ai_name%~  
[%medal_level%>=25]*(cd11:5)欢迎%ai_name%，请多多关照~  
[%guard%]**(cd12:5)欢迎舰长%ai_name%回家！
```

内置100个冷却通道，其中 `0~9` 已被系统使用，用户自定义建议为 `10~99`，应该够用了。

注意：B站连续发送弹幕的冷却时间为1秒，与本程序的弹幕冷却系统无关。

等待通道

针对指定某个通道的弹幕，机器人发送一条后，至少等待他人的多少条弹幕后才会重新发送这条弹幕；自己发送的弹幕不会包含在这“多少条弹幕”之后。

格式：`wait通道:弹幕数量`

通道与冷却通道一样，可选 `0~99`。不同的通道互不影响。

在程序一开始连接上直播间时，所有等待通道的值为 `99`，相当于 `waitX:99`，一般无影响。

如定时消息：

(wait12:3) 某定时发送的弹幕

自动发送一条弹幕后，并且第一条和第二条弹幕之间至少有3条弹幕；不满足条件则不会发送出去。

因为有主播嫌弃机器人话多，专门加的功能。

示例：不连续的定时弹幕

结合条件和选项，设置定时任务，机器人不连续发送两条弹幕：

```
[%wait10% < 1]**** // 没有别人发弹幕时，自己不连续发弹幕
(wait10:0) 喜欢%upname%的小伙伴可以动动小手点个关注~
(wait10:0) 关注主播不迷路,主播带你上高速~
(wait10:0) 喜欢主播的可以点点关注哦~
(wait10:0) 赠送一个B坷拉，领取主播专属勋章哦~
```

通过条件 `%wait10% < 1` 先判断该等待通道能否发送，这样影响的是这一块代码中的所有弹幕；

至于 `(wait10:0)`，则只是标记这些弹幕的等待通道是 10，用于下一次的条件判断。

子账号选择

通过“子账号”系统，可以使用不同的账号来发送弹幕。每个账号都有各自对应的序号，默认登录且绑定捐赠版的为**主账号**，ID=0，其余添加的子账号序号从1开始。

注意：所有账号登录都有时效性，请注意更新或清理失效的账号。

格式：`(account:参数)`，可以简写为`(ac:参数)`。其中参数可以是：

| 参数 | 说明 |
|-----------------------------|---------------------|
| <code>(ac:?)</code> | 在所有子账号中随机使用一个 |
| <code>(ac:??)</code> | 在主账号+所有子账号中随机使用一个 |
| <code>(ac:序号)</code> | 账号的序号，主账号是0，子账号从1开始 |
| <code>(ac:开始序号-结束序号)</code> | 账号的序号范围，在其中随机使用一个 |
| <code>(ac:UID)</code> | 使用指定UID的账号 |

示例：使用子账号

```
[%guard%=1]*(ac:1,cd10:5)欢迎舰长~
欢迎观众~
```

命令操作

有一些自定义的命令，如 `>block(123456)`，如下：

| 命令 | 中文 | 说明 |
|--|--------|---|
| abort() | 终止 | 终止本流程后面弹幕 |
| block(uid, hour) | 禁言 | 禁言用户，uid 可使用参数 %uid% 获得 |
| block(uid) | 禁言 | 同上，默认使用自动禁言的时间 |
| unblock(uid) | 解禁 | 解除禁言 |
| eternalBlock(uid, markname) | 永久禁言 | 永久禁言某用户（需保持程序在线），markname 为标记名字（避免时间长了改名不知道） |
| appointAdmin(uid) | 任命房管 | 任命用户为房管，仅主播可用 |
| dismissAdmin(uid) | 撤销房管 | 取消用户的房管身份，仅主播可用 |
| delay(msecond) | 延时 | 延迟执行后面所有待执行的操作，单位毫秒 |
| addGameUser(listId, uid) | 添加游戏用户 | 添加用户至游戏队列，listId从0到99，使用 %>inGameUsers(listId, uid)%判断在不在。 程序重启后清空 ，长期保存可使用 addGameNumbers |
| addGameUser(uid) | 添加游戏用户 | 同上，默认listId使用0，可使用 [%in_game_users%] 快速判断 |
| removeGameUser(listId, uid) | 移除游戏用户 | 从游戏队列中移除用户 |
| removeGameUser(uid) | 移除游戏用户 | 同上，默认listId使用0 |
| addGameNumber(listId, uid) | 添加游戏数值 | 添加用户至游戏队列，listId从0到99，使用 %>inGameNumbers(listId, uid)%。 |
| addGameNumber(uid) | 添加游戏数值 | 同上，默认listId使用0，可使用 [%in_game_numbers%] 判断 |
| removeGameNumber(listId, uid) | 移除游戏数值 | 从游戏队列中移除数值 |
| removeGameNumber(uid) | 移除游戏数值 | 同上，默认listId使用0 |
| addGameText(listId, uid) | 添加游戏文本 | 添加用户至游戏队列，listId从0到99，使用 %>inGameTexts(listId, uid)%。 |
| addGameText(uid) | 添加游戏文本 | 同上，默认listId使用0，可使用 [%in_game_texts%] 判断 |
| removeGameText(listId, uid) | 移除游戏文本 | 从游戏队列中移除用户 |
| removeGameText(uid) | 移除游戏文本 | 同上，默认listId使用0 |
| showValueTable(caption, key, field1, field2,) | 显示值表格 | 显示多个值的表格，详见下方示例 |
| sendGift(giftId, num) | 赠送礼物 | 赠送礼物，只支持 id 的方式 |
| execRemoteCommand(cmd) | 执行远程命令 | 执行远程控制（见下面） |
| execRemoteCommand(cmd, 0) | 执行远程命令 | 执行远程控制，不发送回馈通知 |
| sendPrivateMsg(uid, msg) | 发送私信 | 向指定用户发送私信 |

| 命令 | 中文 | 说明 |
|--|-------------|---|
| sendRoomMsg(roomId, msg) | 发送直播间弹幕 | 向指定直播间发送弹幕（注意：该操作不参与发送的队列，频繁使用可能导致发送过快） |
| sendRoomEmoji(roomId, emojiId) | 发送直播间表情 | 向指定直播间发送表情（注意：同上） |
| sendEmoji(emojiId) | 发送表情 | 上面命令的省略房号版（注意：同上） |
| showScreenDanmu(text) | 显示全屏弹幕 | 将text作为本地全屏移动的弹幕显示，受相关设置影响，需打开“全屏弹幕” |
| timerShot(msecond, msg) | 延迟发送 | 定时多少 毫秒 后发送弹幕msg（msg允许为另一命令，多个命令使用 %m% 分隔） |
| localNotify(msg) | 本地通知 | 发送本地消息通知（非弹幕，只有自己看得到） |
| localNotify(uid, msg) | 本地通知 | 同上，带用户ID |
| speakText(msg) | 播放语音 | 朗读文本 |
| speakTextSSML(ssml) | 播放语音 SSML | 使用微软语音来播放SSML文本（不能有换行）；需要配置API |
| speakTextUrl(url) | 播放语音 URL | 播放在线合成的语音 |
| setVoiceSpeaker(name) | 设置语音发音人 | 不同语音有不同的发音人，详细看程序说明 |
| setVoiceSpeed(speed) | 设置语音音速 | 0~100 |
| setVoicePitch(pitch) | 设置语音音调 | 0~100 |
| setVoiceVolume(volume) | 设置语音音量 | 0~100 |
| openUrl(url) | 打开网址 | 浏览器打开网址 |
| connectNet(url) | 连接网址 | 用于连接其他API，不管返回结果 |
| getData(url, [callback]) | get数据 | 后台连接网址（GET），返回JSON数据，callback详见“获取网络数据回调”示例。回调中的 %text% 为纯文本结果 |
| postData(url, data, [callback]) | post数据 | 同上（POST） |
| postJson(url, data, [callback]) | postjson | 同上，以JSON格式发送（仅Content-Type区别） |
| postHeaderData(url, headers, data, [callback]) | post头数据 | 同上。headers多个之间用 &分隔，格式：header1=value1&header2=value2&header3=value3 |
| downloadFile(url, path, [callback]) | 下载文件 | 下载网络文件至本地，成功后触发自定义事件。回调中的 %text% 为文件路径 |
| sendToSockets(cmd, data) | 发送至 socket | 发送给所有包含cmd的已连接的WebSocket，如果cmd为空，则发送给所有WebSocket |
| sendToLastSocket(cmd, data) | 发送至最后socket | 发送给最后连上的WebSocket，无视cmd（甚至没收到cmds也行） |
| runCommandLine(cmd) | 运行命令行 | 运行操作系统的命令行 |
| startProgram | 运行程序 | 打开外部程序（例如守护进程） |

| 命令 | 中文 | 说明 |
|---|---------|--|
| setSetting | 设置某项配置 | 等同于v3.7之前的setValue，已不建议使用 |
| removeSetting | 移除某项配置 | 同上 |
| setValue(key, val) | 设置值 | 保存值到配置文件，通过%{key}%获取，重启后仍在。默认保存在“heaps”分组下，使用“group/key”指定分组 |
| addValue(key, val) | 添加值 | 在原先值的基础上，添加val大小 |
| setValues(exp, val) | 批量设置值 | 批量修改 已有的 值，exp为正则表达式。不允许批量设置非默认分组（即不能带“/”） |
| addValues(exp, val) | 批量添加值 | 批量添加已有的值 |
| setValuesIf(exp, [condition], newVal) | 批量设置值如果 | 按条件批量修改已有的值，[condition] 同弹幕条件（带方括号），详见下方“批量修改配置” |
| addValuesIf(exp, [condition], delta) | 批量添加值如果 | 按条件批量添加已有的值 |
| removeValue(key) | 移除值 | 移除配置文件中的单个值 |
| removeValues(exp) | 批量移除值 | 移除配置文件中的多个值（不允许带“/”），exp为正则表达式 |
| removeValuesIf(exp, [condition]) | 批量移除值如果 | 按条件移除配置文件中的多个值 |
| openFile(path) | 打开文件 | 打开文件 |
| playSound(path) | 播放声音 | 播放音频文件 |
| improveMusic(username, order) | 提升点歌 | 点歌提前播放，order为提升的索引值 |
| cutMusic(username) | 切歌 | 切歌，仅限正在播放该用户自己点的歌时 |
| cutMusic() | 切歌 | 立即切歌，无论是谁点的 |
| addMusic(path/name) | 添加音乐 | 添加本地音乐（完整路径）或者自动搜索网络歌曲到播放列表 |
| playMusic() | 播放音乐 | 播放点歌姬的音乐；若没有歌曲，尝试播放下一首 |
| pauseMusic() | 暂停音乐 | 暂停点歌姬的音乐 |
| toggleMusicState() | 切换音乐状态 | 切换音乐的播放/暂停状态 |
| messageBox(text) | 消息弹窗 | 弹窗提示 |
| sendLongText(text) | 发送长文本 | 发送长文本，自动分割成多条 |
| appendFileLine(fileName, format) | 添加文件行 | 追加一行文本保存至“程序目录/dirName/fileName”末尾，支持变量。可用于保存送礼记录、上船记录等 |
| insertFileAnchor(fileName, anchor, content) | 插入文件锚点 | 插入文字至文件中的“anchor”前。可组建HTML的 <table> 内容 |
| writeTextFile(fileName, text) | 保存文本文件 | 写入文本至“程序目录/fileName”，允许相对/绝对路径。注意路径的一些问题，比如“C:\new\a.txt”中的“\n”会被当做换行，可以使用“C://new/a.txt”的形式代替 |
| removeFile(fileName) | 删除文件 | 删除文件“程序目录/fileName” |

| 命令 | 中文 | 说明 |
|---|----------|---|
| removeTextFileLine(fileName, line) | 移除文本文件行 | 删除文本文件中的某一行，行号从1开始 |
| modifyTextFileLine(fileName, line, newText) | 修改文本文件行 | 修改文本文件中的某一行，行号从1开始 |
| fileEachLine(fileName, startLine, code) | 文件每一行 | 读取txt文件的每一行，执行code命令。fileName 可以是相对文件名、绝对文件路径。startLine 为从哪一行开始读，可忽略。code中多行弹幕/代码需要使用 %n% 替换 \n，并且需要使用 \% 来转义替换 %，例如 \%text\%。 \%number\% 获取行号（从1开始）， \%text\% 获取本行内容；会覆盖原有的这两个变量 |
| csvEachLine(fileName, startLine, code) | CSV每一行 | 读取csv文件（表格）的每一行，同上。使用转义的 %\$数字% 来获取这一行中的每列元素，如 \%\$2\% 获取当前行第二列 |
| AIReply(text, maxLen) | AI回复 | 调用AI回复某文字，并直接以弹幕的形式发送。maxLen为长度上限，默认单条弹幕，超出则不回复 |
| AIChat(text, code) | AI聊天 | 调用AI回复文字，若有结果则执行code代码，使用 \%text\% 来获取回复结果。具体调用哪个AI，由设置中的选项决定。如果是ChatGPT，UID固定为0 |
| TXChat(text, code) | 腾讯AI聊天 | 调用腾讯AI闲聊，同上 |
| ChatGPT(uid, text, code) | GPT聊天 | 调用ChatGPT闲聊，同上。如果开启了“AI分析”，那么 %text% 会是对应的msg |
| ignoreWelcome(uid) | 不自动欢迎 | 不自动欢迎某用户 |
| setLocalName(uid, name) | 设置专属昵称 | 设置用户专属昵称 |
| joinBattle(type) | 开启大乱斗 | 开启大乱斗，1普通，2视频 |
| triggerEvent(event) | 触发事件 | 触发自定义事件，可在“事件动作”中响应；附带当前最近处理的数据。若有多个，都会触发 |
| triggerReply(msg) | 触发自动回复 | 触发自动回复（未开启会被忽略），可作为有参数的方法进行传递。若有多个，都会触发；若未找到合适的回复动作，会报一个警告 |
| call(event, arg1, arg2...) | 调用 | 触发事件，可以传递若干个参数，使用 %\$1% 这样的指令来获取参数 |
| getReplyExecutionResult(msg) | 获取回复执行结果 | 获取 第一个 符合 msg 弹幕的回复的执行结果，并返回 最终弹幕 的拼凑结果，使用 %n% 连接（写入到文件时会替换为换行）。该回复的代码中的命令会被立刻执行；若没有相应的回复，返回空字符串 |
| getEventExecutionResult(event) | 获取事件执行结果 | 同上 |
| setTimerTask(id, time) | 开启定时任务 | id为单行注释的文字，格式：//id。time：<1时重置时间，=-1时关闭，=0时切换开关，=1时开启，>1时设置为对应秒数并重置时间（关闭的话不会开启）。不建议设置1秒，因为弹幕冷却2秒 |
| orderSong(songName, uname) | 点歌 | 自动点歌，uname可以为任意字符 |
| addBannedWord(word, anchor) | 添加违禁词 | 在指定" anchor"处插入" word" |

| 命令 | 中文 | 说明 |
|--|---------|--|
| showCSV(filePath) | 显示CSV | 显示表格文件，自动判定编码 |
| simulateKeys(keys) | 模拟按键 | 模拟例如“ctrl+a”等按键，不分大小写。加号使用“add”代替；不支持汉字等非键盘键位字符。可用 0x70 这样的形式模拟Win的F1键等。 |
| simulatePressKeys(keys) | 模拟按下按键 | 同上 |
| simulateReleaseKeys(keys) | 模拟松开按键 | 同上 |
| simulateClick() | 模拟点击 | 模拟鼠标左键点击 |
| simulateClick(x, y) | 模拟点击 | 移动到指定x,y点，再模拟单击 |
| simulateClickButton(flag) | 模拟点击按钮 | 模拟鼠标点击，flag为WIN API 中的mouse_event事件的DWORD参数，即鼠标事件的位或值。如6为左键单击，24为右键单击。可简写为 left、right、middle、x |
| simulateClickButton(flag, x, y) | 模拟点击按钮 | 移动到指定x,y点，再模拟各按键点击；flag同上 |
| moveMouse(x, y) | 移动鼠标 | 移动鼠标到指定x,y点 |
| moveMouse(dx, dy) | 调整鼠标 | 移动鼠标，相对于现在的位置差 |
| execScript(path) | 执行脚本 | 执行放在 程序目录/control/ 文件夹下的bat或者vbs脚本，或绝对路径 |
| copyText(text) | 复制文字 | 把文字复制到剪贴板 |
| setRoomTitle | 设置直播标题 | 主播设置直播间的大标题 |
| setRoomCover(path) | 设置直播封面 | 主播设置直播间的封面图片，path为图片路径（不会裁剪，只是压缩上传） |
| setLocalMode(true/false) | 设置本地模式 | 可通过回复、定时等开关本地模式，不发送弹幕 |
| reconnectRoom() | 重新连接直播间 | 强制重连直播间 |
| connectRoom(roomId) | 连接直播间 | 切换到任意直播间，支持房间号或身份码 |
| SQLExec | SQL执行 | 需要开启数据库。执行SQL语句，如创建表格、插入记录 |
| SQLQuery | SQL查询 | 需要开启数据库。执行SQL语句并显示查询结果，如查找弹幕记录 |
| saveScreenShot(screenId, x, y, w, h, path) | 保存窗口截图 | 保存指定位置的窗口截图为本地图片。screenId:屏幕ID，只有一个屏幕则为0；x/y/w/h:坐标与宽高；path:要保存的文件路径 |
| showWindow(hwnd, state) | 显示窗口 | 将窗口置为指定状态，state: normal/min/max/hide/show |
| sendWindowMessage(hwnd, text) | 发送窗口消息 | 向指定窗口发送文字类的消息，模拟文字输入，如社交软件聊天界面。需要输入框获取到焦点才能输入 |
| moveWindow(hwnd, x, y, w, h) | 移动窗口 | 移动窗口至指定的坐标和尺寸 |
| sendEmail(address, subject, body) | 发送邮件 | 使用设置的账户向指定邮件地址发送邮件 |

在自动回复的每一条弹幕中使用符号 `>` 开头，紧接着 `func(arg...)` 格式，将执行命令，而不发送弹幕（若不是上述命令，将改为弹幕发送）。

注意：若要同时操作多个命令，可用 `\n` 隔开（相当于多条弹幕）

批量修改配置

针对 `setValuesIf(exp, [condition], newVal)`、`removeValuesIf(exp, [condition])` 这两个较为复杂的命令。

`exp` 为正则表达式，将会操作所有key满足该表达式的配置。

当且仅当在 `[condition]` 中：

- `_VALUE_` 替换为当前遍历到的值(value)
- `_$x_` 替换为exp正则捕获的文字，其中 `x=0` 时为键(key)，`x>0` 时为正则的捕获组，等同于自动回复中的 `_%x%`
- `_{key}_` 替换为读取配置文件，等同于正常情况下的 `_{key}_%`

当且仅当在 `newVal` 中：

- `_[exp]_` 替换为计算表达式（不支持外面再嵌套任意内容）

例如：

- 删除7天没来的用户记录：
`removeValuesIf(not_come_(\d+), [_VALUE_>7])`
- 今日没签到的用户重置连续签到为0：
`setValuesIf(signin_keep_(\d+), [!_{signin_today__$1}_], 0)`
- 用户上船天数+1：
`setValuesIf(guard_days_\d+, [1], [_VALUE_+1])`

显示值列表

`showValueTable(caption, key, field1, field2, field3...)`

显示一个对话框，列出所有需要的值。

`caption` 是最顶上的标题，支持变量。

`key` 为用来遍历的带有可变值的正则表达式，例如 `integral_(\d+)`，将会遍历所有使用 `setValue` 存储的键满足这个表达式的数值。其中必须有带括号的捕获组形如 `(\d+)` 作为**变化的源头**，用来影响所有 `field` 的变化；而在各 `field` 中可以使用 `_KEY_` 或者 `_ID_` 来替换为该值。

`field` 允许多个，每个的格式为：`标题:键`，标题是表格的标题（第一行），键是用来读取响应设置的。标题及中间的冒号可忽略。用 `_KEY_` 或 `_ID_` 来替换为表达式中变化源。单个数字类型的 `field` 末尾允许用 `:<` 从小到大、`:>` 从大到小排序，详见下方示例。

在 `key` 和 `field` 中，可使用 `_counts/`、`_heaps/` 前缀来指定读取哪一个配置文件，`key` 默认为 `_heaps` 即用户自定义的配置；`field` 默认跟随 `key` 的配置。

其中，`_counts/` 文件为 `安装目录/danmaku_counts/房号.ini`，自动保存每个直播间的数据，**不同直播间不共享**；`_heaps/` 为 `安装目录/heaps.ini`，**所有直播间共享一套数据**。

示例：显示所有用户积分

显示签到的用户ID、昵称、积分，按积分从大到小排序

在之前的默认签到代码中，未加上保存昵称，所以uname是空的


```
>showValueTable(积分查询, integral_(\d+), ID:"_ID_", 昵称:uname__ID_, 积分:integral__ID_:>)
```

假定某个用户 id 是 123456, 那么通过 `integral_(\d+)` 遍历到 `integral_123456`, 则 `_ID_` 是 123456, 通过这一行来获取后面的 `name_123456`、`integral_123456` 等的值。

示例：显示所有打卡记录

```
>showValueTable(今天共第%[%{daka}%+0]%人打卡, daka_sum_(\d+), ID:"_ID_", 累计:daka_sum__ID_:>, 昵称:uname__ID_, 积分:integral__ID_, 连续:daka_keep__ID_, 本月:daka_month__ID_, 今日:daka_today__ID_)
```

示例：显示用户总金瓜子

```
>showValueTable(氪金列表, _counts/gold/(\d+), ID:"_ID_", 金瓜子:gold/_ID_:>, 昵称:_heaps/uname__ID_)
```

解析JSON：网络数据回调

三个联网命令：`getData`、`postData`、`postJson`, 最后一个参数可带有一个“回调入口”。其实这是一个事件，添加该回调同名的事件，即可获取到联网返回的数据。

目前只支持JSON格式的返回数据，使用 `%.键1.键2.键3%` 这样的格式依次获取JSON对象的值，例如：

```
{
  "data": {
    "room": {
      "roomname": "房间名字"
    },
    "anchor": {
      "uname": "名字",
    }
  }
}
```

其中的roomname获取方式：`%.data.room.roomname%`, uname获取方式：`%.data.anchor.uname%`

注意：如果替换出来的内容**包含换行**，一样会替换到代码中，将会导致弹幕姬出错！

JSON容错处理

V4.0.4版本修改：当JSON对象**没有该键**或者**索引超出数组范围**时，默认还是原字符串（例如 `%.data.list.1000%`，原样发出）

如果要允许不存在的键或者超出范围的索引，可以在后面加上英文字符“?”，如：`%.data.list?.1000?%`，当没有list或者list数量不到1000时，会返回空白文本。“?”只影响单个键或者索引，例如若没有data，会原样返回 `%.data.list?.1000?%`，相当于报错。

这样的作用是，JSON的**某些键可能是从变量中获取**，需要**先解析变量**才能开始读取完整的JSON，而变量可能会出问题。

该版本之前，默认允许错误，返回空白。

示例：获取主播信息

添加命令至回复、事件等任意一项，动作：


```
>getData(https://api.live.bilibili.com/xlive/web-room/v1/index/getInfoByRoom?
room_id=%room_id%, RoomInfoCallBack)
```

再添加一项事件: `RoomInfoCallBack`, 动作:

```
>localNotify(舰长数量: %.data.guard_info.count%, \
    粉丝数量: %.data.anchor_info.relation_info.attention%, \
    粉丝团: %.data.anchor_info.medal_info.fansclub%, \
    主播等级: %.data.anchor_info.live_info.level%)
```

关于多线程安全性

一般情况下, 处理弹幕的命令会有短暂的延迟, 尽管程序已经做了一定的优化, 但两条弹幕前后的操作必定不会连续, 存在 2s 延迟 (受限于平台发送弹幕的最短间隔), 因此如果同时处理多条弹幕, 可能会出现数据丢失问题。

典型情况, 如多人同时打卡, 增加一天打卡次数:

```
观众1: >setvalue(daka, %{daka}+1)%
观众2: >setvalue(daka, %{daka}+1)%
```

假设 `%{daka}% = 0`, 那么 `%[%{daka}+1]% = 1`, 那么执行的命令有可能会变成:

```
观众1: >setvalue(daka, 1)
观众2: >setvalue(daka, 1)
```

虽然有两人打卡, 但结束后 `daka` 的值依旧为 1, 而不是 2。

修改方法: 使用 `>addvalue(daka, 1)` 或者语法糖 `{daka}++` 来替代 `>setvalue(daka, %{daka}+1)%`, 更安全且简洁易懂!

```
观众1: >addvalue(daka, 1) -> 变成1
观众2: >addvalue(daka, 1) -> 变成2
```

简单命令示例

示例: 自动打卡

添加定时任务, 设置时间为 86400 (一天秒数)

添加发送的文本:

```
>sendRoomMsg(直播间房号, 打卡)
```

例如:

```
>sendRoomMsg(123456, 打卡)
```

这个命令会在房号为“123456”的直播间发送弹幕“打卡”。

注意: 需要关闭“仅直播时发送”, 或加上条件 `[%living%+1]` (未开播也执行)

示例: 自定义朗读弹幕格式

默认的设置只支持朗读用户名和弹幕内容，不能自定义朗读的格式。

这样设置自定义朗读的格式，加上“某某说”：

添加回复：(.+)，表示接收所有弹幕（不包括机器人自动发的），

```
>speakText(%uname%说: %text%)
```

注意：需要关闭设置里的朗读弹幕，否则会重复播放 2 遍

示例：禁言小游戏

扣1禁言、关注主播或赠送小心心解除禁言（需要房管或主播）

回复中添加一栏，关键词为 ^1\$，回复：

```
已自动禁言，赠送小心心或关注主播解禁\n>block(%uid%, 1)\n>addGameUser(%uid%)
```

这里执行了三个操作：

1. 回复
2. 禁言
3. 添加到游戏用户，等待解除禁言

答谢 — 感谢送礼中添加：

```
[%in_game_users%,%origin_gift_name%=小心心]***已解除禁言\n>unblock(%uid%)\n>removeGameUser(%uid%)
```

其中 [...]*** 中的星号为优先级，保证数量超过其余弹幕即可。

%in_game_users% 确保是扣 1 被禁言（因为禁言的时候添加到了游戏用户）的，而非被手动禁言的用户

答谢 — 感谢关注中添加：

```
[%in_game_users%]*已解除禁言\n>unblock(%uid%)\n>removeGameUser(%uid%)
```

实测已关注的，先取关再马上重新关注，收不到通知，需要等会儿再关注

在两个答谢中，执行了三个操作：

1. 回复
2. 解除禁言
3. 从游戏用户中移除，后续小心心不再触发该游戏，而是普通的答谢

示例：上船自动私信

答谢 — 感谢送礼中添加：

```
[%guard_buy%,%guard_first%=1]感谢%ai_name%开通%gift_name%!\n>sendPrivateMsg(%uid%, 感谢开通大航海，可加入粉丝群: xxx)
```

%guard_buy% 判断是否是购买舰长的通知；%guard_first% 判断是否第一次上船，第一次=1，续船=0，掉船后重新上船=2。

也可用 %guard_count% 来读取上船的次数作为条件，%guard_count%=0 表示第一次上船。

`%guard_count%` 只计算本程序运行时购买舰长的用户，每次舰长+1，提督+10，总督+100。

后续版本已添加 `FIRST_GUARD` 用户第一次上船事件，可直接使用。

示例：送礼优先点歌

答谢-感谢送礼中添加：

```
[%gift_name%=喵喵]>improveSongOrder(%uname%,5)
```

赠送一个喵喵则提前5首歌播放。可将 5 改为 999 表示无限大，或者用 `%gift_gold% / 1000` 表示每1000金瓜子礼物可提前一首歌，如下：

```
[%gift_name%=喵喵]>improveSongOrder(%username%,[%gift_gold%/1000]%)
```

示例：弹幕切歌

房管可以切所有歌，普通观众只能切自己点的歌

自动回复添加**表达式**：`^切歌$`

设置动作：

```
[%admin%]*>cutOrderSong()  
>cutOrderSong(%uname%)
```

示例：远程开关AI弹幕回复

默认仅机器人和主播可远程控制，而其他人不行。通过此方法设置特定用户（例如舰长、房管）开关。

添加关键词表达式：`^开启AI回复$`

添加回复：

```
>execRemoteCommand(开启弹幕回复,0)\n>setvalue(reply,1)\n已开启AI弹幕回复，发送“关闭”  
结束
```

添加关键词表达式：`^关闭(AI回复)?$`（回复“关闭”两字即可，可不用全打）

添加回复：

```
[%guard%, %reply%]>execRemoteCommand(关闭弹幕回复)\n>setvalue(reply,0)
```

逐一说明：

- `>execRemoteCommand`：运行仅机器人和主播才能用的远程指令，参数2的 0 表示不自动回复（因为有自定义的另一个回复，覆盖掉默认的）
- `\n`：多个指令或多条弹幕，用 `\n` 标记隔开
- `setvalue(reply, 1)`：设置变量 `reply` 的值为1，用来判断是否需要关闭
- `[%guard%, %reply%]`：舰长且 `reply=1`，若是则执行后面的，否则不做操作

示例：远程添加违禁词

添加回复：`添加违禁词\s*(.+)`，动作：

```
[%admin%]>addBannedword(%$1%,|more_ban)\n已设置，再发将自动禁言
```

违禁词格式：

```
^[^不都]*(违禁词1|违禁词2|more_ban)
```

在接收到房管发的形如“添加违禁词 笨蛋”的弹幕时，会在 `more_ban` 的左边插入 `|笨蛋`。

积分查看

添加下述代码到任意可执行的地方，定时/回复/事件 均可，需要查看时点击“发送”按钮。

示例：所有用户积分

```
/// 显示所有用户积分
>showValueTable(积分查询, integral_(\d+), ID:"_ID_", 昵称:uname__ID_, 积分:integral__ID_:>, 动作:action__ID_, 坐骑:mount__ID_)
```

示例：所有打卡记录

```
/// 显示所有打卡记录
>showValueTable(今天共第%[%{daka}%+0]%人打卡, daka_sum_(\d+), ID:"_ID_", 累计:daka_sum__ID_:>, 昵称:uname__ID_, 积分:integral__ID_, 连续:daka_keep__ID_, 本月:daka_month__ID_, 今日:daka_today__ID_)
```

事件动作

收到服务器CMD消息、一些程序的运行状态的改变，都会触发本程序中的“事件”。

部分数据可直接用 `%uid%`、`%uname%` 等变量获得，部分CMD未读取数据，返回为空。

大部分CMD等同于B站后台CMD，也有一些是自创的。可用事件CMD如下：

主程序事件

| 事件 | 说明 |
|-----------------------|---|
| START_UP | 程序启动完成 |
| START_WORK | 开播后启动程序，或者程序启动后再开播 |
| LOGIN_FINISHED | 连接直播间并登录账号后 |
| SHUT_DOWN | 程序关闭的一瞬间 |
| REMOTE_BLOCK | 远程禁言，信息为最后一条弹幕 |
| REMOTE_BLOCK_OVERRIDE | 同上，但是会覆盖掉自带的禁言回复 |
| FIND_USER_BY_UNAME | <code>>unameToUid(昵称)%</code> 的结果，可能是送礼或弹幕数据 |
| SOCKET_MSG_RECEIVED | 接收到网页程序的socket消息 |
| PLAY_SOUND_FINISHED | <code>>playSound(path)</code> 播放音频文件结束， <code>%text%</code> 获取文件路径 |

示例：远程禁言回复

禁言 xxx 通过倒找弹幕发送人昵称的方法，通过弹幕禁言用户。其中所有属性同 DANMU_MSG，例如 %uid%、%uname% 等。如果禁言对象是房管，那么将会禁言失败。

添加事件 REMOTE_BLOCK_OVERRIDE，添加动作：

```
[%uid%=%my_uid%]**>因为太帅无法被禁言
[%uid%=%up_uid%]**>已禁言主播（狗头保命）
[%admin%]**>无法禁言房管
>已禁言：%uname%
```

注意：添加本事件，将会屏蔽系统自带的禁言回复

时间事件

| 事件 | 说明 |
|-----------------|----------------------|
| NEW_HOUR | 每小时0分触发 |
| NEW_DAY | 每天0点0分触发 |
| NEW_DAY_FIRST | 每天0点或每天第一次启动 |
| DAY_END | 每天23:59:30（剩半分钟处理任务） |
| NEW_MONTH | 每月1号0点 |
| NEW_MONTH_FIRST | 每月1号0点或每月第一次启动 |
| MONTH_END | 每月最后一天23:59:30 |
| NEW_YEAR | 每一年开始的0点 |
| NEW_YEAR_FIRST | 每年开始0点或每年第一次启动 |
| YEAR_END | 每年最后一天23:59:30 |
| NEW_WEEK | 每周一0点 |
| NEW_WEEK_FIRST | 每周一0点或每周第一次打开 |
| WEEK_END | 每周结束 |

如果想指定间隔准点发送，例如每15分钟，可在整点时开启时长为900秒的定时任务

NEW_DAY 和 NEW_DAY_FIRST 的异同：

同：

- 每天0点都会触发

异：

- NEW_DAY：如果神奇弹幕没开，那么第二天启动时需要手动点事件中对应的“发送”按钮
- NEW_DAY_FIRST：启动时会自动执行新的一天的操作

注意：v3.7.4版本之前，NEW_DAY_FIRST 有 bug，每次启动都会触发

示例：打卡计数

发送“签到”或“打卡”进行打卡，并回复第几个以及累计几天；每人每天只能打一次卡。

本处只做一个简单计数，建议使用下方的“高级打卡”示例

添加自动回复: `^(签到|打卡)$`, 动作:

```
[%{daka_today_%uid%}]*>您已打过卡
[%living%+1]>打卡成功，您是今天第[%{daka}%+1]个，累计[%{daka_sum_%uid%}%+1]天\n
>setvalue(daka, [%{daka}%+1])\n
>setvalue(daka_today_%uid%, 1)\n
>setvalue(daka_sum_%uid%, [%{daka_sum_%uid%}%+1])
```

添加事件: NEW_DAY, 动作:

```
[%living%+1]>removeValues(daka_today_\d+)\n>setValue(daka, 0)
```

`[%living%+1]` 是用来保证即使开启了“仅在直播时回复”也能发送回复的弹幕

示例：高级打卡

在以上的打卡计数中，添加了：打卡查询、连续天数、每月累计天数、满足天数报告。

因数量较多，建议直接复制以下代码，使用“工具-菜单”中的“粘贴代码片段”一键添加。

```
[
{
    "anchor_key": "神奇弹幕:AutoReply",
    "enabled": true,
    "key": "^((签到|打卡)\\s*(\\(\\(|\\..*|\\d+)?)?$",
    "reply": "/// 记录每位用户是第几个打卡以及累计天数\n[%{daka_today_%uid}%]*>您已打过卡\n[%living%+1]>签到成功，您是今天第[%{daka}%+1]%个，本月[%{daka_month_%uid}%+1]%天\n\\\n\t>setValue(daka, [%{daka}%+1]%\n\\\n\t>setValue(uname_%uid%, %uname%\n\\\n\t>setValue(daka_today_%uid%, 1)\n\\\n\t>setValue(daka_sum_%uid%, [%{daka_sum_%uid}%+1]%\n\\\n\t>setValue(daka_month_%uid%, [%{daka_month_%uid}%+1]%\n\\\n\t>setValue(daka_keep_%uid%, [%{daka_keep_%uid}%+1]%\n\\\n\t>setValue(integral_%uid%, [%{integral_%uid}%+1000]%\n\\\n\t>triggerEvent(DAKA_MONTH_%{daka_month_%uid}%+1)%)"
},
{
    "anchor_key": "神奇弹幕:AutoReply",
    "enabled": true,
    "key": "^((查询(打卡|签到)|(打卡|签到)查询)$",
    "reply": "[%living%+1]连续[%{daka_keep_%uid}%]%天，本月[%{daka_month_%uid}%]%天，累计[%{daka_sum_%uid}%]%天"
},
{
    "action": "/// 重置每天打卡的人数\n[%living%+1]>setValue(daka, 0)\n\\\n // 重置今日打卡人数\n\t>removeValuesIf(^daka_keep_(\\d+)\$, [!_{daka_today__\$1_}])\n\\\n // 未连续签到断开\n\t>removeValues(daka_today_\\d+)\n// 重置每人是否打卡",
    "anchor_key": "神奇弹幕:EventAction",
    "enabled": true,
    "event": "NEW_DAY_FIRST"
},
]
```

```
    "action": "/// 重置每月打卡天数\n[%living%+1]>setValues(daka_month_\\d+,
0)",
    "anchor_key": "神奇弹幕:EventAction",
    "enabled": true,
    "event": "NEW_MONTH_FIRST"
  },
  {
    "action": "您已打卡满21天，可找主播领小礼物~",
    "anchor_key": "神奇弹幕:EventAction",
    "enabled": true,
    "event": "DAKA_MONTH_21"
  }
]
```

点歌姬事件

%uname% 点歌的用户， %text% 歌曲名字

| 事件 | 说明 |
|-----------------------------|---|
| ORDER_SONG_SUCCEED | 点歌成功， %text% 获取歌名， %prev_time% 获取几秒钟后播放， %first% 获取第几首播放， %.addBy% 获取点歌用户昵称 |
| ORDER_SONG_SUCCEED_OVERRIDE | 同上，但是会覆盖掉自带的点歌成功回复 |
| ORDER_SONG_PLAY | 点歌开始播放， %text% 歌名， %uname% 点歌的用户名， %come_time% 点歌时间戳(秒) |
| ORDER_SONG_NOT_FOUND | 点歌未找到， %text% 获取搜索词 |
| ORDER_SONG_FREQUENCY | 点歌过于频繁 |
| ORDER_SONG_NO_MEDAL | 点歌未戴勋章 |

| 事件 | 说明 |
|-------------------------|---|
| ORDER_SONG_NO_COPYRIGHT | 点歌的所有音源都无法播放时触发 |
| ORDER_SONG_COPY | 点歌已复制歌名 |
| ORDER_SONG_IMPROVED | 播放顺序提前， <code>%number%</code> 获取当前顺序，从1开始，1为下一首 |
| ORDER_SONG_CUTTED | 被手动切歌，即调用 <code>>cutOrderSong</code> |
| SONG_PLAY_FINISHED | 歌曲播放结束事件， <code>%.addBy%</code> 不为空表示是点歌 |
| CURRENT_SONG_CHANGED | 歌曲开始播放(包括手动播放的)或播放列表结束。 |
| ORDER_SONG_BLOCKED | 点歌由于关键词被阻止， <code>%{.key}%</code> 获取关键词 |

可用歌名 `[%text%=""]` 判断歌曲是否有效，`[%uname%=""]` 判断是在点歌姬中手动搜索添加的歌曲还是弹幕点歌的歌曲。

示例：点歌提示未带勋章

添加事件：`ORDER_SONG_NO_MEDAL`，动作：

```
(cd35:600)请戴粉丝牌点歌
```

加了**冷却通道**，最多十分钟提醒一次。

示例：点歌后立即切歌

在播放随机音乐时，用户点歌则立即播放。若已在播放其他用户点的歌，则无效。

添加点歌成功事件：`ORDER_SONG_SUCCEED`，动作：

```
[%playing_song% != "", "%song_order_uname%" = ""]>cutOrderSong()
```

弹幕事件

其中加粗事件表示程序中默认已用上（允许重复使用），并且能获得详细用户ID、昵称、粉丝勋章等信息。

未加粗事件则只是接收数据，未进行解析。

| 事件 | 说明 |
|-------------------------------|------------------------|
| LIVE | 开播【欢迎语】 |
| PREPARING | 下播【结束语】 |
| ROOM_CHANGE | 房间信息改变 |
| ROOM_RANK | 排名改变 |
| DANMU_MSG | 接收到弹幕【自动回复】 |
| SEND_GIFT | 有人送礼【答谢送礼】 |
| WELCOME_GUARD | 舰长进入（不会触发） |
| ENTRY_EFFECT | 舰长、高能榜、老爷进入【欢迎舰长】 |
| WELCOME | 老爷进入 |
| INTERACT_WORD | 用户进入【欢迎】 |
| ATTENTION | 用户关注【答谢关注】 |
| SHARE | 用户分享直播间 |
| SPECIAL_ATTENTION | 特别关注直播间，可用 %special%判断 |
| ROOM_REAL_TIME_MESSAGE_UPDATE | 粉丝数量改变 |
| SUPER_CHAT_MESSAGE | 醒目留言 |
| SUPER_CHAT_MESSAGE_JPN | 醒目留言日文翻译 |
| SUPER_CHAT_MESSAGE_DELETE | 删除醒目留言 |
| SPECIAL_GIFT | 节奏风暴/定制的专属礼物 |
| ROOM_BLOCK_MSG | 用户被禁言， %uname% 昵称 |
| GUARD_BUY | 有人上船 |
| FIRST_GUARD | 用户初次上船 |

| 事件 | 说明 |
|----------------------------|--|
| NEW_GUARD_COUNT | 船员数量改变事件， <code>%uname%</code> 新船员昵称， <code>%num%</code> 获取大航海数量，附带直播间信息json数据 |
| USER_TOAST_MSG | 上船附带的通知 |
| HOT_RANK_CHANGED | 热门榜排名改变 |
| HOT_RANK_SETTLEMENT | 荣登热门榜topX |
| HOT_RANK | 热门榜xx榜topX， <code>%text%</code> 获取排名 |
| ONLINE_RANK_V2 | 礼物榜（高能榜）刷新 |
| ONLINE_RANK_TOP3 | 高能榜TOP3改变 |
| ONLINE_RANK_COUNT | 高能榜改变 |
| NOTICE_MSG | 上船等带的通知 |
| COMBO_SEND | 礼物连击 |
| ANCHOR_LOT_CHECKSTATUS | 天选时刻前的审核 |
| ANCHOR_LOT_START | 开启天选 |
| ANCHOR_LOT_END | 天选结束 |
| ANCHOR_LOT_AWARD | 天选结果推送 |
| VOICE_JOIN_ROOM_COUNT_INFO | 申请连麦队列变化 |
| VOICE_JOIN_LIST | 连麦申请、取消连麦申请 |
| VOICE_JOIN_STATUS | 开始连麦、结束连麦 |
| WARNING | 被警告， <code>%text%</code> 可获取内容 |
| CUT_OFF | 被超管切断 |
| room_admin_entrance | 设置房管 |
| ROOM_ADMINS | 房管数量改变 |
| MEDAL_UPGRADE | 勋章升级，仅送礼物后触发，需设置中开启“监听勋章升级”。 <code>%medal_level%</code> 获取新等级（但用户当前勋章不一定是本直播间） |

考虑到发送弹幕的限制，如果同一个事件添加多条弹幕，那么会按队列顺序发送，而不会一口气发完。

并且有冷却（默认2秒左右），可以在开头加上冷却通道例如 `(Cd78:0)` 消除冷却

另外，需注意 `PREPARING`，若开启了“仅直播时回复”，那么大部分动作（包括弹幕、命令等）将不会执行，可在下播结束语中操作

示例：复读机

指定某一位用户的弹幕，重复其弹幕

添加事件：DANMU_MSG，动作：

```
[%uname%=用户昵称]%text%
```

示例：QQ群推送开播消息

支持可通过网络访问的任意机器人接口。

以酷推为例：<https://cp.xuthus.cc>，按其说明配置

添加开播事件：LIVE，动作：

```
>connectNet(https://push.xuthus.cc/group/[skey]?c=[开播消息])
```

其中 [skey] 为您的酷推Skey，[开播消息] 按服务格式自定义

一切配置妥当，开播时将会自动发送消息至QQ群。

示例：下播自动关机

下播30秒后电脑自动关机

添加下播事件：PREPARING，动作：

```
>timerShot(30000, >runCommandLine(shutdown -s -t 30))
```

添加开播事件：LIVE，动作：

```
>runCommandLine(shutdown -a)
```

多了延时30秒和响应开播事件，是排除主播意外掉线的情况。

示例：感谢分享直播间

添加分享事件：SHARE

添加弹幕动作：

```
感谢%ai_name%分享%upname%的直播间！
```

其中 %upname% 可以使用菜单中的“自定义变量”来统一设置，也可以固定写死。

示例：上船声音提示

添加事件：GUARD_BUY，动作：

```
>playSound(%app_path%/audios/guard.mp3)
```

有人上船则自动播放 安装目录/audios/guard.mp3，本程序不自带，需要自己找音频文件放上去。也可以是安装目录之外的绝对路径。

示例：上船自动设置房管

需要主播登录，添加事件：FIRST_GUARD，动作：

```
>postData(https://api.live.bilibili.com/live_user/v1/RoomAdmin/add,
admin=%uid%&anchor_id=%room_id%&csrf_token=%csrf%&csrf=%csrf%&visit_id=)
```

示例：成为第x位船长

事件中添加事件：NEW_GUARD_COUNT，动作：

```
恭喜%uname%成为第%number%位船员！
```

示例：恭喜勋章升级

需要在设置中开启“监听勋章升级”，否则不会一直刷新勋章。

送完礼物后勋章升级（仅送礼，**不包括**上船、每天第一条弹幕）

添加事件 MEDAL_UPGRADE，动作：

```
[%medal_level%>=5, %anchor_room_id%=%room_id%] 恭喜%ai_name%勋章升级
至%medal_level%~
```

注意：用户此时佩戴的粉丝勋章（即 %anchor_room_id%）不一定是当前直播间的，所以要判断一遍。

示例：天选时刻中奖播报

添加事件：ANCHOR_LOT_AWARD，动作：

```
恭喜%uname%中奖：%text%
```

示例：保存弹幕/送礼记录

保存弹幕记录到“mydir/danmu.txt”：

事件：DANMU_MSG

动作：>appendFileLine(mydir/danmu.txt, %uname%: %text%)

保存送礼记录到“mydir/gift.txt”：

事件：SEND_GIFT

动作：>appendFileLine(mydir/gift.txt, %gift_name% %number% %uname%)

大乱斗事件

| 事件 | 说明 |
|------------------------------|--|
| PK_BATTLE_PRE | 大乱斗准备，10秒后开始 |
| PK_BATTLE_SETTLE | |
| PK_BATTLE_START | 大乱斗开始 |
| PK_BATTLE_PROCESS | 大乱斗双方送礼 |
| PK_ENDING | 大乱斗尾声，最后几秒 |
| PK_BATTLE_END | 大乱斗结束 |
| PK_END | %level%判断胜负，-1输，0平，1赢，可用来计算连胜 |
| PK_BATTLE_SETTLE_USER | |
| PK_BATTLE_SETTLE_V2 | |
| PK_LOTTERY_START | 大乱斗胜利后的抽奖 |
| PK_BEST_UNAME | PK最佳助攻，%uname%昵称；%level%:1赢,0平,-1输；%gift_coin%总积分(=金瓜子/100) |
| CALL_ON_OPPOSITE | 本直播间的观众跑去对面串门 |
| ATTENTION_OPPOSITE | 本直播间观众关注了对面主播 |
| SHARE_OPPOSITE | 本直播间观众分享了对面直播间 |
| ATTENTION_ON_OPPOSITE | 对面观众关注了本直播间 |
| PK_MATCH_INFO | 获取对面直播间信息，详见“大乱斗匹配信息”示例，%gift_coin%获取高能榜总积分(=金瓜子数/100)，%number%获取高能榜人数 |
| PK_MATCH_ONLINE_GUARD | 获取对面直播间舰长在线人数，详见“对面在线舰长播报”示例 |
| PK_WINNING_STREAK | 大乱斗连胜事件，%number%获取次数，至少为2 |

示例：大乱斗结束前提醒

大乱斗结束前30秒提醒：一次大乱斗为5分钟，开始后4分半发送弹幕，270秒=270000毫秒。

添加事件：PK_BATTLE_START，动作：

```
>timerShot(270000, 离大乱斗结束还有30秒)
```

示例：大乱斗赠积分卡

添加事件：PK_BATTLE_PRE，动作：

```
>setValue(pk_ceng, 0)
```

添加事件：SEND_GIFT，动作：

```
[%pking%, {%pk_ceng}%=0, %gift_gold% >= 100000, %gift_num%=1]>setvalue(pk_ceng, 1)\n>sendGift(20004, 1)
```

添加事件: `PK_BATTLE_END`, 动作:

```
[{%pk_ceng}%=0]>setvalue(pk_ceng, 1)\n>postData(https://api.live.bilibili.com/xlive/lottery-interface/v2/pk/join, id=%pk_id%&roomid=%room_id%&type=pk&csrf_token=%csrf%&csrf=%csrf%&visit_id=)
```

示例: 大乱斗最佳助攻

添加事件: `PK_BEST_UNAME`, 动作:

```
[%level%>0, %gift_coin% >= 100]感谢本场最佳助攻: %uname%
```

仅当赢了, 并且本次累计送礼有超过100积分(10000金瓜子)才感谢

示例: 大乱斗尊严票

对面超过100积分(1万金瓜子)而自己还是0积分的时候, 送一个吃瓜保尊严`(`_`)_`。

添加大乱斗即将结束事件: `PK_ENDING`

此事件的时间为大乱斗结束前3秒左右。

添加动作:

```
[%pk_my_votes%=0, %pk_match_votes%>100]>sendGift(20004, 1)
```

20004 为吃瓜的礼物ID, 1为数量

示例: 自动开启大乱斗

每次结束后5秒, 自动重新开始大乱斗的匹配。

事件: `PK_BATTLE_SETTLE`, 动作:

```
>timerShot(5000, >joinBattle(1))
```

其中 1 为普通大乱斗, 2 为视频大乱斗。

仅支持 obs, 使用直播姬的主播无效。

第一次大乱斗需手动点“发送”开启。

示例: 主播互相串门

对面主播过来串门时, 自己也过去发弹幕。

添加事件: `INTERACT_WORD`, 动作:

```
[%pking%, %uid%=%pk_uid%]>sendRoomMsg(%pk_room_id%, 我也来串门啦~)
```

示例: 大乱斗匹配信息

添加事件: `PK_MATCH_INFO`, 将在大乱斗开始的时候, 获取对面直播间的信息。

可参考 https://api.live.bilibili.com/xlive/web-room/v1/index/getInfoByRoom?room_id=您的房间ID 的结果，其 `data` 中的值。

例如：

- 主播名称： `%.anchor_info.base_info.username%`
- 勋章名称： `%.anchor_info.medal_info.medal_name%`
- 舰长数量： `%.guard_info.count%`
- 粉丝数量： `%.anchor_info.relation_info.attention%`
- 粉丝团： `%.anchor_info.medal_info.fansclub%`
- 主播等级： `%.anchor_info.live_info.level%`
- 主播积分： `%.anchor_info.live_info.score%`
- 房间名称： `%.room_info.title%`
- 分区名称： `%.room_info.area_name%`

示例动作：

```
匹配到：%.anchor_info.base_info.username%, %.guard_info.count%舰长,
%.anchor_info.relation_info.attention%粉丝
```

示例：大乱斗连胜（旧）

v3.7.4 添加此事件

添加事件： `PK_END`，动作：

```
[%level%>0]>setvalue(winCount, [%{winCount}%+1%])\n>triggerEvent(winCountReport)
[%level%<0]>setvalue(winCount, 0)
```

添加事件： `winCountReport`，动作：

```
[%{winCount}%>=3] 目前连胜场数： %{winCount}%
```

如果要手动调整连胜数量（例如5），可以添加以下动作，手动点击“发送”，事件可以留空：

```
>setvalue(winCount, 5)
```

每天清零，事件： `NEW_DAY_FIRST`，动作：

```
>setvalue(pk_win, 0)
```

V4.2.0 支持连胜事件 `PK_WINNING_STREAK` 了， `%number%` 直接获取到连胜数量

示例：大乱斗连胜

添加事件： `PK_WINNING_STREAK`，添加回复：

```
[%number% > 5] 恭喜达到%number%连胜！
```

示例：对面在线舰长播报

添加事件： `PK_MATCH_ONLINE_GUARD`，将在大乱斗开始的时候，获取对面直播间在线舰长人数。

`%number%` 获取总人数, `%.guard1%` 总督, `%.guard2%` 提督, `%.guard3%` 舰长

动作:

```
[%number%>30]*危! 对面有%number%名舰长在线!  
[%number%>20, %.guard1%>0]**对面有[%number%-%.guard1%]名舰长在线, %.guard1%名总督  
[%number%>0]对面有%number%名舰长在线!
```

PK对面事件

这是对面对直播间的事件, 均需要开启大乱斗消息同步。

| 事件 | 说明 |
|------------------|-----------|
| PK_DANMU_MSG | 对面的弹幕消息 |
| PK_SEND_GIFT | 对面的礼物消息 |
| PK_INTERACT_WORD | 对面的用户进入 |
| PK_ATTENTION | 对面新增关注 |
| PK_SHARE | 对面有人分享直播间 |

私信事件

| 事件命令 | 说明 |
|---------------------|-------|
| RECEIVE_PRIVATE_MSG | 接收到私信 |

需要在设置中开启“接收私信”, 每当有未读的消息发过来时触发该事件。可获取以下常用值:

| 代码 | 说明 |
|-----------------------------------|----------------------------|
| <code>%uname%</code> | 发送者用户名 |
| <code>%uid% / %.talker_id%</code> | 发送者UID |
| <code>%text%</code> | 发送内容。若有换行以“%n%”分隔 |
| <code>%.last_msg.msg_type%</code> | 消息类型, 例如 1 为用户私信, 10 为系统通知 |

示例: 自动回复私信

通过命令 `>sendPrivateMsg(%uid%, 回复内容)` 来发送私信, 实现聊天与自动回复。

以下示例实现了当私信包含“hello”时, 显示消息到弹幕姬上, 并自动回复“你好”:

```
[%.last_msg.msg_type% != 1]***  
[[[%text% ~ hello]]]>localNotify(收到私信: %username%: %text%)\n  
>sendPrivateMsg(%uid%, 你好~)
```

示例: 私信闲聊机器人

通过智能闲聊的接口来实现私信的机器人智能回复功能。

添加私信事件: `RECEIVE_PRIVATE_MSG`:


```
[%.last_msg.msg_type% != 1]***  
>aiChat(%text%, >sendPrivateMsg(%uid%, 【自动回复】\%text\%))
```

当收到私信时，机器人会自动回复。

示例：解析私信内容

通过用户发过来的私信，将其中的关键信息保存到本地。

例如用户发送“设置我的地址：浙江省杭州市xxxx”，神奇弹幕永久保存该数据，用以后续发货表单自动填写。

添加私信事件 `RECEIVE_PRIVATE_MSG`：

```
[%.last_msg.msg_type% != 1]***    // 忽略系统通知等无关消息  
[[[%text% ~ ^设置我的]]]>triggerReply(%text%)  
[[[%text% ~ ^兑换奖品]]]>triggerReply(%text%)  
[[[%text% ~ ^我的信息$]]]>sendPrivateMsg(%uid%, 名字: %{info/name_%uid%}%, 地址: %  
{info/addr_%uid%}%, 号码: %{info/num_%uid%}%)
```

添加回复 `^设置我的(.+)[: :](.+) (^ 在正则表达式中为开头的意思)：`

```
[[[%$1% = 名字]]]>setValue(info/name_%uid%, %$2%)\n>sendPrivateMsg(%uid%, 已设置您  
的名字为: %$2%)  
[[[%$1% = 地址]]]>setValue(info/addr_%uid%, %$2%)\n>sendPrivateMsg(%uid%, 已设置您  
的地址为: %$2%)  
[[[%$1% = 号码]]]>setValue(info/num_%uid%, %$2%)\n>sendPrivateMsg(%uid%, 已设置您  
的号码为: %$2%)
```

示例：自动生成Excel表单

本程序并不直接支持 Excel，所有的相关操作都需要导入/导出为 CSV 文件（本质上就是 txt 文件）。

通过添加一行数据，将用户操作内容、个人信息保存到表格文件中。

例如用户发送“兑换奖品：红包”，结合上一个示例的设置信息，可能生成以下表格格式（标题需要在使用前手动加上）：

| UID | 用户名 | 奖品 | 收件人 | 手机号 | 收件地址 | 时间 |
|--------|-----|----|-----|-------------|----------------|---------------------|
| 123456 | 用户1 | 手作 | 某人 | 12345678901 | 浙江省杭州市 xxxx | 2022-02-02 02:02 |
| 654321 | 用户2 | 红包 | 某某 | | | 2022-02-22 22:22 |

添加回复 `^兑换奖品[: :](.+) $：`

```
[[["%{info/name_%uid}%" = ""]]*>sendPrivateMsg(%uid%, "请先发送“设置我的名字: xxx”
来设置收件人姓名")
[[["%{info/num_%uid}%" = ""]]*>sendPrivateMsg(%uid%, "请先发送“设置我的号码: xxx”
来设置手机号码")
[[["%{info/addr_%uid}%" = ""]]*>sendPrivateMsg(%uid%, "请先发送“设置我的地址: xxx”
来设置收件地址")
[[[%$1% = 手作; %$1% = 红包]]>appendFileLine(兑奖.csv, \
    %uid%, %uname%, %$1%, %{info/name_%uid}%, %{info/num_%uid}%, %
    {info/addr_%uid}%, %>time(yyyy-MM-dd hh:mm)%)\n\
    >sendPrivateMsg(%uid%, 操作成功)
```

生成的 `兑奖.csv` 文件可直接使用 Excel 打开、修改，或另存为 Excel 表格文件。

如果生成的文件在www文件夹中，则可以通过浏览器访问，非常适合挂在服务器上供他人下载。

示例：批量发送Excel信息

Excel 菜单中可导出为 CSV 文件，去除所有格式，只保留数据，每列之间以英文逗号“,”隔开

以奖品发货为例，用户私信自动添加到表格（也可以是上船/礼物/弹幕/中奖等任意事件），主播手动发货后，使用神奇弹幕批量回复“已发货”：

继上述表格文件，任意处添加代码：

```
>csvEachLine(兑奖.csv, >sendPrivateMsg(\%$1%, 【发货提醒】您的“%$3%”已发货
~)%n%>delay(3000))
```

从表格中读取 UID（第一列使用 `%$1%` 读取，又因为是在 `>csvEachLine` 中调用，加上反斜杠 `\` 转义）和奖品（第三列），向相应的用户发送私信。

可多加一列快递单号，手动填写，通过私信一并发送。

延时 `>delay(3000)` 表示发送私信之间间隔 3 秒，瞬间大量发送私信会导致被封禁。

原始数据

私信的 Json 数据，可通过 `%.key1.key2.key3%` 这样的形式来获取。另外，通过接口附加上了“sender”字段，可获取发送者的信息（API不稳定，不一定获取到）。具体示例如下（省略了部分不常用数据）：

```
{
  "talker_id": 324495090, // 用户ID
  "session_type": 1,
  "at_seqno": 0,
  "top_ts": 0,
  "group_name": "",
  "group_cover": "",
  "is_follow": 1,
  "is_dnd": 0,
  "ack_seqno": 193772296863792,
  "ack_ts": 1647245795072485,
  "session_ts": 1647309089148477,
  "unread_count": 6,
  // 最后一条消息。例如获取消息类型: %.last_msg.msg_type%
  "last_msg": {
    "sender_uid": 324495090,
    "receiver_type": 1,
    "receiver_id": 20285041,
```

```
"msg_type": 1,      // 消息类型, 1 用户私信
"content": "{\\"content\\":\\"测试消息\\"}",
"msg_seqno": 194835318046736,
"timestamp": 1647309089,
"at_uids": null,
"msg_key": 7075138663693631915,
"msg_status": 0,
"notify_code": ""
},
"group_type": 0,
"can_fold": 0,
"status": 0,
"max_seqno": 194835318046736,
"new_push_msg": 0,
"setting": 0,
"is_guardian": 0,
"is_intercept": 0,
"is_trust": 0,
"system_msg_type": 0,
"live_status": 0,
"biz_msg_unread_count": 0,
// 发送者信息 (视网络, 可能获取不到)
"sender": {
  "code": 0,
  "message": "0",
  "ttl": 1,
  "data": {
    "mid": 324495090,
    "name": "般若菠萝包rollora",
    "sex": "女",
    "face":
"http://i0.hdslb.com/bfs/face/63340c49aa7915426f4ba5019cd27a5cbec098b8.jpg",
    "face_nft": 0,
    "sign": "自由人, 生活向主播, 经纪人",
    "rank": 10000,
    "level": 5,
    "jointime": 0,
    "moral": 0,
    "silence": 0,
    "coins": 0,
    "fans_badge": true,
    "fans_medal": {
      "show": false,
      "wear": false,
      "medal": null
    },
  },
  "is_followed": true,
  "live_room": {
    "roomStatus": 1,
    "liveStatus": 0,
    "url": "https://live.bilibili.com/11046586?
broadcast_type=0&is_room_feed=1",
    "title": "房间标题",
    "cover":
"http://i0.hdslb.com/bfs/live/new_room_cover/5987df8547248533f87bcbb40926742d8ce
07a17.jpg",
    "roomid": 11046586,
    "roundStatus": 0,
```

```
        "broadcast_type": 0,
        "watched_show": {
            "switch": true,
            "num": 220,
            "text_small": "220",
            "text_large": "220人看过",
            "icon":
"https://i0.hdslb.com/bfs/live/a725a9e61242ef44d764ac911691a7ce07f36c1d.png",
            "icon_location": "",
            "icon_web":
"https://i0.hdslb.com/bfs/live/8d9d0f33ef8bf6f308742752d13dd0df731df19c.png"
        },
    },
}
```

过滤器事件

这一类事件专门用于在某些操作之前进行过滤，相当于高度自定义的黑白名单。

当结果的弹幕包含命令 `>reject()` 时，将不会执行正在过滤的操作。

若有多行，同其他代码一样，随机获取一行；其他弹幕或命令正常发送。

| 过滤事件 | 说明（reject时） |
|--------------------------|--|
| FILTER_MUSIC_ORDER | 阻止点歌 |
| FILTER_AUTO_REPLY | 阻止所有的自动回复 |
| FILTER_DANMAKU_MSG | 阻止弹幕消息显示在弹幕姬上 |
| FILTER_DANMAKU_COME | 阻止进入消息显示在弹幕姬上 |
| FILTER_DANMAKU_GIFT | 阻止礼物/上船消息显示在弹幕姬上 |
| FILTER_DANMAKU_ATTENTION | 阻止关注消息显示在弹幕姬上 |
| FILTER_DANMAKU_NOTICE | 阻止普通通知(Common Notice)显示在弹幕姬上 |
| FILTER_KEYWORD_BLOCK | 阻止关键词自动禁言 |
| FILTER_API_EVENT | 阻止HTTP触发事件的接口， <code>%text%</code> 获取事件名字 |
| FILTER_AI_REPLY | 阻止触发AI回复， <code>%text%</code> 获取弹幕。但是在弹幕姬菜单手动点击“AI回复”会进行强制回复。 避免不必要的回复可节约token消耗，提升性能 |

| 过滤事件 | 说明 (reject时) |
|---------------------|---|
| FILTER_AI_REPLY_MSG | 阻止AI回复发送弹幕， <code>%text%</code> 获取弹幕， <code>%reply%</code> 获取回复内容 |

若有多个相同过滤器，只要**任意其中一个**包括 `>reject()` 命令，则本操作不允许。

另外，过滤器不支持发送选项（冷却通道、等待通道）；若有需要，放到条件判断中。

注意：默认开启过滤器，可能会在一定程度上造成卡顿，可在弹幕设置中关闭。

示例：点歌过滤

只允许舰长或者带本房间粉丝勋章的用户点歌

事件： `FILTER_MUSIC_ORDER`

```
["%anchor_room_id%" != "%room_id%", !%guard%]>reject()
```

两个条件：`不是舰长`、`未戴本房间勋章`，当成立时，执行 `>reject()`，即**阻止**点歌，当做无事发生。

示例：只显示舰长进入

只显示舰长进入的消息，其他用户进入都不显示

事件： `FILTER_DANMAKU_COME`

```
[!%guard%]>reject()
```

示例：过滤免费礼物

赠送的小心心、辣条等不会显示在弹幕姬上

事件： `FILTER_DANMAKU_GIFT`

```
[%gift_gold% = 0]>reject()
```

示例：自定义过滤器

相关函数： `filterReject(filter)`

在事件中，事件为自定义的**过滤器名字**，即参数中的 `filter`。相应动作带有 `>reject()` 则表示**拒绝**，返回1；若不拒绝，则**默认通过**，返回0。

除了 `>reject()`，过滤器中的其余操作都会如普通弹幕命令一样**正常执行**。



☒ 启用

发送

my_filter

[%uname% ~ 懒一夕]>reject()

☒ 启用

发送

DANMU_MSG

[%>filterReject(my_filter)%]>localNotify(被过滤器阻止)
[!%>filterReject(my_filter)%]>localNotify(过滤器允许通过)



[主播] 懒一夕智能科技官方 111

被过滤器阻止



示例：自定义词库过滤

相关函数：inFilterList(filter, content)

返回的弹幕改为词库，多个词语之间用空格或者换行分隔。

只要词库中其中一个词语在 content 中，则返回 1。否则返回 0。



☒ 启用

发送

词库1

词语1 词语2 词语3

☒ 启用

发送

词库2

敏感词1 敏感词2

☐ 启用

发送

事件命令

[%>inFilterList(词库1, 敏感词1)%]*>localNotify(在词库1中)
[%>inFilterList(词库2, 敏感词1)%]*>localNotify(在词库2中)
[!%>inFilterList(词库1, 敏感词1)%]>localNotify(不在词库中)

在词库2中

回车发送弹幕



示例：自定义正则过滤

相关函数：inFilterMatch(filter, content)

返回的弹幕改为正则表达式，多个表达式用多行表示。

只要 content 满足其中一个表达式，则返回 1。否则返回 0。

定时

回复

事件

远程

禁言

☒ 启用

发送

词库1

词语1|词语2|词语3

☒ 启用

发送

词库2

敏感词\d+

☐ 启用

发送

事件命令

[%>inFilterMatch(词库1, 敏感词1)%]*>localNotify(在词库1中)

[%>inFilterMatch(词库2, 敏感词1)%]*>localNotify(在词库2中)

[!%>inFilterMatch(词库1, 敏感词1)%]>localNotify(不在词库中)

在词库2中

在词库2中

不在词库中

在词库2中

回车发送弹幕

录播事件

| 事件命令 | 说明 |
|-----------------------|---|
| LIVE_RECORD_SAVED | 录播结束，%file_path% 获取视频文件的绝对路径，%file_name% 获取文件名（不包含后缀） |
| LIVE_RECORD_FORMATTED | 开启“保存为mp4”后，转换格式成功时触发此事件，%file_path% %file_name% 获取信息 |

警告事件

| 事件命令 | 说明 |
|---------------|--|
| NOT_FIND_USER | 未找到 <code>%>unameToUid(昵称)%</code> 的用户, <code>%text%</code> 获取原昵称 |

服务端事件

| 事件命令 | 说明 |
|--------------------|--|
| NEW_WEBSOCKET | 新连接进入 (此时尚未收到它的cmds) |
| WEBSOCKET_CMD5 | socket设置对应类型, <code>%text%</code> 获取, 英文逗号分隔, 例如: "SONG_LIST,DANMAKU" |
| SOCKET_MSG_RECEIVE | Socket接收到消息的事件 (去除一些内置cmd) |
| SOCKET:[cmd] | Socket接收到对应cmd消息的事件 (所有cmd) |

示例：远程计数

例如送指定礼物做10个蹲起, 这时候需要显示蹲起的数量。

添加新连接事件: `WEBSOCKET_CMD5`, 动作:

```
[%text%=SQUAT]>sendToSockets(SQUAT, {"cmd":"SQUAT", "data":%{squat}%})
```

添加 `初始化蹲起数量` 的动作 (事件建议留空, 手动点击发送按钮):

```
>setValue(squat, 0)\n>sendToSockets(SQUAT, {"cmd":"SQUAT", "data":0})
```

添加 `蹲起数量+1` 的动作 (事件自选, 可空着手动点发送来增加次数):

```
>setValue(squat, %[ %{squat}%+1%])\n\n>sendToSockets(SQUAT, {"cmd":"SQUAT", "data":%[ %{squat}%+1%])
```

创建 `www/squat.html` 文件, 写入:

```
<head>
  <title>神奇弹幕蹲起</title>
  <script src="js/jquery.js"></script>

  <style type="text/css">
    .card {
      background-color: #fff;
      box-shadow: 0 4px 8px 0 rgba(0, 0, 0, 0.2);
      transition: 0.3s;
      /* width: 40%; */
      border-radius: 5px;
      padding-left: 20px;
      padding-right: 20px;
    }

    .card:hover {
      box-shadow: 0 8px 16px 0 rgba(0, 0, 0, 0.2);
    }
  </style>
```

```

</head>

<body>
  <div class="card" style="display:inline-block">
    <h1 id='count'>蹲起个数: 0</h1>
  </div>

  <script type="text/javascript">
    $(document).ready(function () {
      var ws = new WebSocket("ws://__DOMAIN__:__WS_PORT__");
      ws.onopen = function () {
        ws.send({'cmd':"cmds", "data":["SQUAT"]}');
      };
      ws.onmessage = function (e) {
        var json = JSON.parse(e.data);
        var cmd = json['cmd'];
        switch (cmd) {
          case 'SQUAT':
            console.log(json);
            var count = parseInt(json['data']); // 这就是个数
            $("#count").html('蹲起个数: ' + count);
            break;
        }
      };
    });
  </script>
</body>

```

访问 `localhost:5520/squat.html` (域名端口按照设置的来), 显示 蹲起数量: 0。

每次点事件中“蹲起数量+1”那一项的“发送”按钮, 对应蹲起数量加一。

全局快捷键事件

事件中, 以 `KEY:` 开头, 则会自动注册全局快捷键, 例如 `KEY:ctrl+1`, 则是注册快捷键 `ctrl+1`, 全局皆有效。此时任意地方按下 `ctrl+1` 就会执行这个事件里面的代码。

注: 受限于 B 站弹幕发送必须有间隔 (2秒), 代码会有延迟, 不一定是立即执行

Cron 定时事件

事件使用 `TIME:` 开头, 后面使用 Cron 表达式, 即可以实现按具体时间的定时或者间隔操作。

本程序实现了支持标准的 6 字段 Cron 表达式:

秒 分 时 日 月 周

支持的功能包括:

- 通配符 `*`
- 范围 `1-5`
- 步长 `*/5` 或 `1-30/5`
- 列表 `1,3,5`
- 组合 `1-5,10-15/2`

常用示例:

- `0 0 * * * *` - 每小时开始执行

- `*/10 * * * * *` - 每10秒执行
- `0 0 8-10 * * *` - 每天8、9、10点执行
- `0 0/30 8-10 * * *` - 每天8:00, 8:30, 9:00, 9:30, 10:00执行

多语言支持

JS语言

本程序的代码框中均支持 `JavaScript` 编程语言的语法。

支持的 JS 特性依赖于 Qt 版本，当前发布的版本为 Qt5.x，对 JS 的支持概况如下：

- **核心标准：**基于 `ECMAScript 5 (ES5)`，并部分支持 `ES6 (ES2015)` 的特性。
- **支持的关键特性：**
 - **ES5 特性：**严格模式 (`"use strict"`)、`JSON` 对象、数组迭代方法（如 `map`、`filter`）。
 - **部分 ES6 特性：**箭头函数、`let/const`、模板字符串、增强的对象字面量。
 - **不支持的 ES6+ 特性：**类 (`class`)、模块 (`import/export`)、`Promise`、`Proxy`、生成器 (`function*`) 等。

使用方式：在代码框中使用 `js:` 开头，后面的代码就会视作 JavaScript 代码来执行，返回值（`return` 一个字符串）作为发送的字符串，或者要执行的命令。自动回复等所需数据的变量名为 `danmaku`。

JS语法示例

```
js:
// 打印日志（会在弹幕姬+日志文件显示出来，仅限string类型，其他类型需先转换）
console.log(danmaku.getNickname() + ":" + danmaku.getText())
// return的内容将会作为弹幕发送，或者神奇弹幕中支持的可执行命令
// 如果return了空字符串，或者不进行return，那么不会回复弹幕
if (danmaku.getText() == "测试弹幕")
    return "收到测试弹幕";
else
    return "";
```

如果要在 JS 代码中使用环境变量，可以将开头的 `js:` 改为 `var:js:`，会先替换变量，之后再运行 JS 代码。但如果在 JS 代码中再需要使用变量，则会替换为固定的值。

本质上是定义了一个 function，其返回值就是要发送的弹幕，所以直接 `return 弹幕` 即可。

弹幕接口列表

默认变量 `danmaku` 的所有参数如下（按消息类型，很多值都不一定有，务必先输出看看）：

其中字符串返回的是 `QString` 类型，这里简写为 `String` 以便观看（毕竟只要知道是字符串就好了）

```
int getMsgType() // 消息类型
long long getUid() // UID，长数字类型
String getNickname()
String getUname() // Nickname和Uname是完全一样的，避免用错
String getUnameColor()
String getText()
```

```

String getTextColor()
DateTime getTimeline()
int isAdmin() // 是否是房管
int getGuard() // 0无, 1总督, 2提督, 3舰长
int isVip()
int issvip()
int isUidentity()
int getLevel() // 用户等级
int getWealthLevel() // 财富等级
int getGiftId() // 礼物ID
String getGiftName() // 礼物名字
int getNumber() // 礼物数量
long long getTotalCoin() // 瓜子数量（旧单位, 1000金瓜子=10电池=1元）
bool isFirst() // 多用途。第一次上船、点歌等待的歌数量（例如下一首播放就是1）、高能榜排名、礼物连击的第一个礼物
long long getReplyMid() // @用户的回复
String getReplyUname()
String getReplyUnameColor()
bool isReplyMystery()
int getReplyTypeEnum()
String getAIReply() // AI回复消息
String getFaceUrl() // 头像URL

```

其中 `MsgType` 用来判断是什么消息，枚举类型如下：

| 值 | 描述 |
|----|-------|
| 1 | 弹幕 |
| 2 | 送礼 |
| 3 | 欢迎 |
| 4 | 点歌 |
| 5 | 上船 |
| 6 | 欢迎舰长 |
| 7 | 粉丝 |
| 8 | 关注 |
| 9 | 禁言 |
| 10 | 普通消息 |
| 11 | 分享直播间 |
| 12 | PK最佳 |
| 13 | 醒目留言 |
| 14 | 其他 |

处理配置文件

默认嵌入了应用配置 `settings` 和用户变量 `heaps`，使用 `read` 和 `write` 函数来读写相应的数据。

settings 为应用配置，请谨慎修改；修改之后应用也不会立即读取。

heaps 为用户配置，随意读写。默认读写的分组为 `heaps`，即 `write("test")` 等同于 `write("heaps/test")`，但 `write("my/test")` 属于另一个分组 `my` 下。

示例：

```
js:
heaps.write("test", "222")
console.log(heaps.read("test"))
```

网络连接

由于嵌入的 JS 引擎并不支持许多原生 JS 模块，网络连接与原生 JS 略有不同。

```
// GET 请求
let response = get("https://api.example.com/data");

// POST 请求
let response = post("https://api.example.com/submit", JSON.stringify({key:
"value"}));

// 使用 fetch 进行更复杂的请求
let response = fetch("https://api.example.com/data", {
  method: "POST",
  headers: {
    "Content-Type": "application/json"
  },
  body: JSON.stringify({key: "value"})
});
```

示例：获取直播间标题

```
js:
let response = get("https://api.live.bilibili.com/room/v1/Room/get_info?
room_id=1");
let data = JSON.parse(response);
console.log(data.data.title)
```

Lua语言

同 JS，代码前缀为 `Lua:`。建议使用前者，与本应用结合得更好。

使用的版本为 Lua 5.2.4。

Lua语法示例

```
lua:
-- 打印信息，会在弹幕姬显示出来，仅限字符串类型
print(danmaku.getNickname() .. ": " .. danmaku.getText())
-- 逻辑判断
if danmaku.getText() == "111" then
  return "is 111"
else
  return "not 111"
end
```

处理配置文件

因为 Lua 使用了第三方库，仅支持标准类型的读写，且无法重载，`read` 和 `write` 必须指明类型，均添加了相应后缀，例如 `readString`、`writeInt` 等等。

示例：

```
lua:
heaps:writeString("test", "111")
print(tostring(heaps:readInt("test")))

if heaps:contains("test") then
    local value = heaps:readString("test")
    print(value)
end
```

Python语言

前缀为 `python`（允许简写为 `py`）、`python3`，分别调用对应的命令。

需要自己本地安装 Python 环境，并且命令行可以直接调用。

传递的 JSON 格式：

```
{
  "danmaku":
  {
    "uid": [xxx],
    "nickname": "[xxx]",
    "text": "[xxx]",
    ...
  },
  "settings_path": "[应用数据路径]/settings.ini",
  "heaps_path": "[应用数据路径]/heaps.ini"
}
```

`应用数据路径` 如果是绿色版，那么就是安装路径；如果是单文件版，那么按照系统设置会不同。

默认全局python不支持第三方库，例如网络库requests，请参考下方虚拟环境。

Python示例

与 js 和 lua 不同的是，使用 `print()` 的输出作为返回的弹幕。

同时并不支持调试，无法打印一些数据，但是可以使用 `print(">localNotify(要打印的文字)")` 来使文字显示在弹幕姬上，而非输出弹幕。

```
python:
import sys
import json

json_data_str = sys.argv[1]

# 这个是弹幕JSON
danmaku = json.loads(json_data_str)["danmaku"]
name = danmaku["nickname"]
```

```

text = danmaku["text"]

# 在这里使用传入弹幕数据进行你的 Python 逻辑
danmu = f"{name}: {text}"

# 返回要发送的弹幕内容
print(danmu)

```

完整的代码，带有一些报错，但不是很有必要。

```

python3:
import sys
import json

if __name__ == "__main__":
    if len(sys.argv) > 1:
        try:
            json_data_str = sys.argv[1]

            # 这个是弹幕JSON
            danmaku = json.loads(json_data_str)["danmaku"]
            name = danmaku["nickname"]
            text = danmaku["text"]

            # 在这里使用传入弹幕数据进行你的 Python 逻辑
            danmu = f"{name}: {text}"

            # 返回要发送的弹幕内容
            print(danmu)

        except json.JSONDecodeError:
            print("Error: Invalid JSON data received.")
        except Exception as e:
            print(f"Python script error: {e}")
        else:
            print("Error: No data received from C++.")

```

处理配置文件

通过本地命令行的方式将弹幕的 JSON 传入 Python，不同于前面的 JavaScript 和 Lua 是直接嵌入到应用程序中，也因此无法直接共享配置文件等信息，而是通过 ini 配置文件来修改，并同步回去。

```

python3:
import sys
import json
import configparser

json_data_str = sys.argv[1]

# 这个是JSON中的配置文件路径
data = json.loads(json_data_str)
settings_path = data["settings_path"] # 应用配置
heaps_path = data["heaps_path"] # 用户配置

# 读取配置文件
config = configparser.ConfigParser()

```

```

try:
    # 以 heaps 为例, 不建议修改 settings
    config.read(heaps_path)
    # 读取配置
    test_value = config.get("General", "test", fallback="default_value")
    print(">localNotify(" + test_value + ")")

    # 修改或添加配置
    if not config.has_section("General"):
        config.add_section("General")
    config.set("General", "test2", "value222") # 分组, key, value

    # 写入配置文件
    with open(heaps_path, 'w') as configfile:
        config.write(configfile)
except configparser.Error as e:
    print(f"Error Settings INI file: {e}")

```

使用Python虚拟环境

通过Python的虚拟环境, 可以为插件安装单独版本的第三方库, 与其他项目不冲突。

```

cd [神奇弹幕的codes文件夹]

# 创建虚拟环境
python -m venv venv

# 2. 激活虚拟环境
# windows:
venv\scripts\activate
# Linux/macOS:
source venv/bin/activate

# 3. 在虚拟环境中安装包
pip install package-name

# 4. 退出虚拟环境
deactivate

```

指定命令行exe

支持使用命令行运行的程序, 调用形式如 `python [文件路径]` 这样的命令 (暂不支持带其他参数)。

使用语法规则 `exec:[exe路径]:` 开头来指定要调用的命令, 以指定虚拟环境的 Python 为例 (Mac 上) :

```

exec:/Users/iwxyi/Library/Application Support/LanYixi/神奇弹幕/codes/venv/bin/python:
import sys
print(">localNotify(111111)")

```

由于路径过长, 以及给大家确定一个通用路径, 添加了统一的变量替换 `<codes>` 来指定为 `数据路径/codes` 文件夹, 例如上述代码可以写为:


```
exec:<codes>/venv/bin/python:
import sys
print(">localNotify(111111)")
```

从代码文件导入

在代码框中，使用 `<file:文件名>` 的标签，该标签会被替换为 `文件名` 表示的代码文件。

其中 `文件名` 可以是：

- 程序 `数据路径/codes/` 文件夹下的代码文件，可以是带后缀，亦或是不带后缀。
- 本地绝对路径 / 相对路径的代码文件
- 以 `http(s)://` 开头的网络代码文件（用以配置同步）

代码文件示例

签到打卡，回复的表达式：`^(签到|打卡)$`，

执行的代码使用 `js:` 开头，表示执行 JS 代码；代码从文件中读取：

```
js:
<file:打卡>
```

`数据路径/codes/打卡.js` 文件内容：

```
if (settings.read("daka_today_" + danmaku.getUid()) == 1)
    return "您已打过卡";

// 开始打卡
count_all = settings.read("daka");
if (count_all == null)
    count_all = 0;
count_all = count_all + 1;
settings.write("daka", count_all);
settings.write("daka_today_" + danmaku.getUid(), 1);
return "打卡成功，您是今天第" + count_all + "位打卡用户";
```

代码文件中使用变量

这需要在主程序的代码框中添加前缀 `var:js:`：

```
var:js:
<file:打卡>
```

这样会执行变量替换操作，`打卡.js` 就可以写作：

```
if (settings.read("daka_today_%uid%") == 1)
    return "您已打过卡";
// ... 剩下的打卡代码 ...
```

`"daka_today_%uid%"` 相比原先的 `"daka_today_" + danmaku.getUid()` 就简短了许多。

Web开发接口

浏览器访问

以下皆需开启设置中的**网络服务**方可生效。

程序目录下的 `www` 文件夹为网站根目录，所有 `html` 文件可放入该位置，浏览器地址栏输入 `域名:端口`（默认为 `localhost:5520`）即可访问，效果同 `Apache` 一致。

Socket通讯

支持 JavaScript 的 WebSocket 通讯，示例如下：

```
var ws = new WebSocket("ws://__DOMAIN__:__WS_PORT__");
ws.onopen = function() {
    ws.send('{"cmd":"cmds", "data":["SONG_LIST"]}');
};
```

其中，`__DOMAIN__:__WS_PORT__` 会自动替换为用户设置的 `域名:端口+1`，若未设置，则默认为 `localhost:5521`。

在 `onopen` 方法中，必须向服务端发送格式为 `{"cmd": "cmds", "data": [需要接收的cmd列表]}` 的 JSON 数据，其中 `cmd列表` 为接收服务端的哪些命令的字符串数组（相当于白名单），可用**事件**中的部分命令。

另外，有专用的特殊 CMD，如下：

- `SONG_LIST`：点歌列表
- `LYRIC_LIST`：歌词
- `CURRENT_SONG`：正在播放的歌曲
- `LIVE_ALL_GIFTS`：本场直播所有礼物（同一人同一礼物合并），仅接受一次

通过WebSocket，在网页中可接收几乎所有的弹幕消息，显示各种动画特效。

为什么要先发一个白名单 `cmds`：程序运行过程中的命令很多，socket 可能也很多，如果每个命令给每个 socket 都发一遍，会非常影响性能

主程序发送消息

发送给网页 WS 使用，使用 `>sendToSockets(cmd, data)` 命令。示例如下：

```
>sendToSockets(SOME_CMD, \
    {\
        "action": "SOME_ACTION",\
        "data": {"key1": "value1", "key2": 222}\
    })
```

第一个参数的 `SOME_CMD` 为 `cmd` 字符串用于**过滤 socket**，即为 `onopen` 发送过来的 `cmds` 中的某一类型才会接收，不会把接收弹幕消息发到只接收礼物消息的 socket 对象上。

第二个参数是一个 JSON 字符串，即网页端接收到的 `data`，具体格式要和网页一致。

接收主程序消息

以点歌姬的点歌列表为例，接收到 `SONG_LIST` 并显示在 DOM 节点中：

```
<ol id="songs" class="live numbers"> </ol> <!-- 被修改的ul -->
```

```
ws.onmessage = function(e) {
    console.log(e.data); // 打印接收的json格式，类型太多，就不一一说明了
    var json = JSON.parse(e.data);
    var cmd = json['cmd'];
    switch (cmd) {
        case 'SONG_LIST':
            var songs = json['data']; // 歌曲对象组成的数组
            var olHtml = '';
            for (var i = 0; i < songs.length; i++) {
                var song = songs[i]; // 歌曲对象，可获取歌名、歌手、用户等
                olHtml += '<li>' + song['name'] + "</li>";
            }
            $("#songs").html(olHtml);
            break;
    }
};
```

注意：本处代码中有个 `cmd` 变量，这个 `cmd` 不是 `cmds` 命令中 `sendToSockets` 的参数一（不过恰好也是 `"SONG_LIST"`），而是参数二的 `action` 中的元素，相当于白名单。

主程序接收消息

通用消息接收事件

在 Web 端向服务端（神奇弹幕主程序，以下统称“主程序”）发送 socket 数据，除了一些内置 CMD 类型外，都会触发 `SOCKET_MSG_RECEIVE` 事件，通过解析 json 的方式（如 `%.cmd%`），可获取其中的信息，进行一系列的操作。

内置 cmd，即[持久化配置](#)、[反向控制主程序](#)中的 cmd，不会触发 `SOCKET_MSG_RECEIVE` 事件。简单来说，就是只响应用户自定义的 socket 消息。

`SOCKET_MSG_RECEIVE` 事件的代码示例：

```
[%.cmd% == ONE_CMD] 执行ONE_CMD动作
[%.cmd% == ANOTHER_CMD] 执行ANOTHER_CMD动作
```

独立cmd接收事件

如果 json 中带有 cmd，则会额外触发 `SOCKET:cmd` 事件，例如：

```
{
  "cmd": "CLICKED",
  "data": {
    "name": "小明"
  }
}
```

会触发 `SOCKET:CLICKED` 事件，可在事件中响应收到的 json，如 `%.data.name%`。

与上面的 `SOCKET_MSG_RECEIVE` 相比，这个可以将不同的 cmd 响应放到不同的代码块中，更加简洁明了。

持久化配置

简单使用可选择 localStorage，但不容易跨设备传输与自动备份。

用户在网页程序中自定义配置，永久保存在安装路径下的 `ext_settings.ini` 中。使用相应的 cmd 实现功能，无需“解锁安全限制”。

- `SET_CONFIG`：在网页中发送该 cmd 的 json，将会保存配置。
`group` 为保存的分组，读取时传入该参数可获取该分组下的所有配置。建议一个应用的配置都保存到同一个分组下。

```
{
  "cmd": "SET_CONFIG",
  "group": "test",
  "data": {
    "key1": 123,
    "key2": "某个值",
    "key3": 10.08,
    "key4": false
  }
}
```

该命令无返回。

- `GET_CONFIG`：返回同样的 cmd，data 中会包含配置的键值对

```
{
  "cmd": "GET_CONFIG",
  "group": "test",
  "data": ["key1", "key2", "key3", "key4", "key5"]
}
```

可以不添加 `data` 或留空 `"data": []`，会返回该 group 下的所有配置。

若是 JS 并且使用了默认的 `magical_danmaku.js`，添加 `function readConfig(data)`，其中的形参 `data` 就是返回的 JSON 的 data，可直接读取里面的数值。

以上面的 `SET_CONFIG` 为例，可能会返回 JSON：

```
{
  "cmd": "GET_CONFIG",
  "data": {
    "key1": 123,
    "key2": "某个值",
    "key3": 10.08,
    "key4": false,
    "key5": null
  }
}
```

读取主程序信息

通过 ws 向主程序发送 `GET_INFO` 格式的 cmd，则会通过内置解释器将一些变量输出为响应的真实数值（都是字符串格式），并按发送的格式返回：

```
{
  "cmd": "GET_INFO",
  "data": {
    "key1": "key1表达式",
    "key2": "key2表达式"
  }
}
```

添加 `function readInfo(data)`，其中形参 `data` 就是返回的 JSON 中的 `data`。

以获取房间ID、标题为例：

```
function socketInitd() {
  var json = {
    cmd: "GET_INFO",
    data: {
      room_id: "%room_id%",
      title : "%room_name%",
      pk      : "%pking%"
    }
  }
  appws.send(JSON.stringify(json));
}

function readInfo(data) {
  roomId = data['room_id']; // 获取到的房间ID
  title  = data['title'];   // 房间标题
  pking  = data['pking'];   // 是否正在PK
}
```

可能返回的数据：

```
{
  "cmd": "GET_INFO",
  "data": {
    "room_id": "123456",
    "title": "标题",
    "pk": "0"
  }
}
```

反向控制主程序

在上述“主程序接收消息”的基础上，指定 CMD 类型，可反向控制主程序。一部分 CMD 需要在扩展中开启 `扩展-网络服务-解锁安全限制` 方有效（默认关闭）。

JSON 格式：

```
{
  "cmd": cmd类型,
  "data": 数据
}
```

服务端可接收 `cmd类型` 如下（不分大小写）：

下述 cmd，都会跳过 `SOCKET_MSG_RECEIVE` 事件。前三项在上面都有详细说明，且无需解锁安全限制。

- `cmds`
- `get_config`
- `set_config`
- `forward`：将 `data` 中的数据发送给其他 socket，`data` 中要同样再包含一层 `cmd` 和 `data`。完整 JSON 如下：

```
{
  "cmd": "forward",
  "data": {
    "cmd": "[转发的命令]",
    "data": "[转发数据]"
  }
}
```

- `set_value`：修改主程序的原始配置，重启主程序生效。可以有两种形式：

```
// 单独修改一项设置
{
  "cmd": "set_value",
  "data": {
    "key": "[key]",
    "value": [value] // 可以是字符串，也可以是整数
  }
}
```

```
// v4.7.2新增，同时修改多项设置
{
  "cmd": "set_value",
  "data": {
    "key1": [value1],
    "key2": [value2],
    "key3": [value3]
  }
}
```

- `send_msg`：使主程序发送弹幕，`data` 为弹幕字符串，使用 `\\n` 来分隔多条弹幕；**不会解析变量**，而是直接发送出去

```
{
  "cmd": "send_msg",
  "data": "要发送的弹幕1\\n弹幕2\\n弹幕3"
}
```

- `send_variant_msg`：使主程序发送带变量的弹幕或命令，例如 `%{key}%`；需要用户信息的例如 `%uid%` 均不可使用。

```
{
  "cmd": "send_variant_msg",
  "data": "数量: %{count}%"
}
```

便携接口

实现了一些简单的接口，便于使用：

- 用户头像（已失效）：`/api/header?uid=123456`，可直接用于 `` 标签

```

```

- 网络代理：`/api/netProxy?url=网址`，可解决本地请求无法跨域问题；其中 网址 若带有参数，则建议进行 **URL 编码**。请求时使用 GET、POST 等不同方法、设置不同 content-type，都会响应转移到代理中，并自动设置当前登录账号的 Cookies。

```
$.ajax({
  url: "http://__DOMAIN__:__PORT__/api/netProxy?url=http://api.live.bilibili.com/xlive/web-room/v1/giftPanel/giftConfig?platform=pc",
  async: false,
  dataType: "JSON",
  success: function (data) {
    giftList = data["data"]["list"];
  }
});
```

该接口除了 `url` 之外的参数都会设置到请求的 header 中，如：`/api/netProxy?url=xxx&referer=xxx&content-type=xxx`

- 触发事件：`/api/event?event=事件名&data=url编码(json)`，会触发指定事件如 `SEND_GIFT`。若使用 post 方式，data 部分可直接使用 json 来发送而不需要 url 编码。
- 配置文件相关：（下列接口需要打开“解锁安全限制”开关）
 - 获取所有配置：`/api/readConfig`，返回 `settings.ini` 文件内容。如果带有参数 `format=json` 则以 JSON 格式返回，否则返回纯 INI 文本；如果带有参数 `key` 则读取指定配置，返回纯字符串
 - 设置单个值：`/api/setConfig?key=xxx&value=xxx&reload=1`，当 `reload` 不为空时自动重新加载所有配置
 - 主程序重新加载所有配置：`/api/reloadConfig`

默认WS工具

在 "www/js" 文件夹下内置了 `magical_danmaku.js`，使用之后，可以更简单地连接主程序的 WebSocket，您只需要关注 Web 应用的实现。

```
<script src="../js/magical_danmaku.js"></script>
```

以下是一些示例：

```
// 全局变量，直接覆盖即可
// 需要使用的 cmd 类型，可以是多个，为空则表示所有（影响性能）
```

```

usedMsgType = ["DANMU_MSG", "SEND_GIFT"];
// 解析CMD，自动调用
function parseCmd(cmd, data) {
    if (cmd == "DANMU_MSG") { /* ... */ }
}

// 配置分组，建议每个小程序一个分组
configGroup = "danmu";
// 如果分组非空，连接后自动获取一次该分组下的所有配置
function readConfig(data) {
    if (data['count'] == 3) { /* ... */ }
}

// WebSocket连接成功后的自定义操作，可以获取一些数据
function socketInited() {
    // 这里是获取房间号 %room_id% 的值，放在键 current_room_id 中
    var json = {
        cmd: "GET_INFO",
        data: { current_room_id: "%room_id%" }
    }
    appws.send(JSON.stringify(json));
}
// 返回上面需要获取的数据，自动调用
function readInfo(data) {
    currentRoomId = data['current_room_id'];
}

```

网页程序打包

所有的网页小程序都是以文件夹的形式放在 `www` 中，每一套小程序（可以是多个网页）对应 `www` 中一个文件夹，并带有 `package.json` 文件（旧版为 `info.json`），可被神奇弹幕识别，添加到“网络服务”的“已安装扩展”中，并添加一些快捷按钮。

以点歌姬为例：

```

{
    "name": "点歌姬",          // 扩展整体名字
    "min_version": "4.4.0",   // 能用的神奇弹幕最低版本，过低会报警告
    "author": "小义",        // 作者名字
    "list": [                 // 允许多个网页，list中一项一个
        {
            "name": "弹幕点歌列表",      // 这是显示出来的名字
            "url": "index.html", // 相对于主机地址的URL路径
            "desc": "显示弹幕点歌的实时列表，播放完毕后自动移除", // 简单描述与说明
            "author": "作者名", // （可空）单独的作者名字
            "css": "list.css", // （可空）便于用户修改的CSS文件相对于www文件夹的路径
            "css_custom": "list_custom.css", // （可空）用户自定义CSS，修改后会覆盖上面一项
            "cover": "cover.png", // （可空）扩展封面图片，默认为 cover.png
            "code": [], // （可空）有些弹幕交互程序需要添加的代码，代码块菜单“复制+继续复制”后粘贴到此处
            "config": "config.html", // （可空）扩展的配置页面
            "homepage": "http://xxx", // （可空）主页链接
            "contact": "http://xxx", // （可空）联系作者
            "reward": "http://xxx", // （可空）打赏链接
        }
    ]
}

```



```

        "dir": "images", // (可空) 打开某一文件夹，比如图片资源，用于用
户替换
        "file": "result.txt", // (可空) 打开某一文件，比如保存的抽奖结果
        "menus": [ // (可空) 自定义扩展的右键菜单
            {
                "name": "自定义菜单",
                "url": "http://xxx",
                "code": "" // 点击这个菜单项要执行的代码/弹幕
            }
        ]
    },
    {
        "name": "当前歌曲名字",
        "url": "/music/playing.html",
        "css": "/music/playing.css",
        "desc": "点歌姬正在播放的歌曲的名字"
    },
    /* ... */
]
}

```

QA

为什么不发送弹幕？

先检查有没有登录、是否在直播或关闭“仅直播时回复”、送免费礼物不答谢，再看下面的方法。

调试模式

以“欢迎”为例，`菜单-调试-调试模式`，开启后每次有人进来都会尝试欢迎，有以下几种情况：

- **什么提示都没有**，甚至看不到人进来：
 - 该用户关闭了进入直播间通知
 - **短时间反复进入直播间都不会有消息**，静等十分钟后重试
- 正常发送欢迎弹幕
- [冷却中]：还在CD，避免刷屏
- **[空弹幕，已跳过]**：有如下两种可能：
 - 按条件跳过（如未戴勋章），即形如 `[xxx]****` 的格式
 - 没有符合条件的回复

空弹幕时，可以打开 `菜单-调试-最后一次候选弹幕`（手速要快，不然会被新弹幕覆盖），查看变量被替换的内容，以及弹幕随机发送的候选项，检查逻辑条件是否有误。

如果自己的直播间不行，开启调试中的“本地模式”（不会真正发送弹幕或送礼物），连接到其他人的直播间再进行测试。

一直在获取房间信息？

需要安装VC_redist: https://aka.ms/vs/15/release/vc_redist.x64.exe 后重启神奇弹幕。

大部分Windows自带，但是不排除一些精简版系统将它去掉了。

点歌没有反应

点歌成功但是播放不了，需要安装解码器（如 LAV），然后重启神奇弹幕。

点歌没有加到歌单，请看下面逐一排查：

1. 检查点歌设置页面的**左上角总开关**是否开启
2. 要保持**播放器页面一直打开**（“召唤点歌姬”）
3. 检查**点歌表达式**是否正确，默认格式：点歌 歌名，无需修改
 - 如果格式正确，则点歌姬会自动出现搜索歌曲的列表
4. 查看**点歌设置页面底部**有没有点歌列表（最下面方块，点右上角的按钮）
 - 如果没有这个列表，则表示点歌的格式不对
5. 检查点歌条件，比如勋章限制、连续点歌
6. 检查事件中的代码，点歌过滤器中设置了拒绝点歌
7. 查看音乐播放器页面的**点歌列表**有没有未播放歌曲
 - 手动暂停的话，点歌后依旧保持暂停状态，是不会继续播放的哦
8. 检查是否加入了点歌**黑名单**（点歌设置页面的右上角）

实时视频不能播放？

需要安装 **LAV解码器**，可直接去网上搜索下载，可能需要重启电脑生效。

讯飞语音没有声音？

同上，需要安装解码器。

怎么添加弹幕窗口到OBS？

扩展-网络服务，选择已安装的“弹幕”扩展。

数据重复显示怎么办？

同一个程序请勿重复打开，否则保存的数据（如弹幕累计数量、礼物累计总金额）、数据库记录（如弹幕历史）会重复插入。

如果想要多开，请复制文件夹再运行。

怎么提bug？

加QQ群：**1038738410**，主动点（" > _ < ; "）

问题/bug/建议，一定要**详细详细详细**！最好来一篇300字小作文！

一些反例：

- 能针对关键词，回复指定弹幕吗？（能看到这句话为什么看不到上面这么多回复示例）
- 为什么运行一段时间后闪退了？（不说详细操作，不可能找到原因）
- 欢迎/回复/点歌/切歌/...不管用？（先看看上面“为什么不发弹幕”再问吧）

参考资料

- B站API列表：<https://github.com/SocialSisterYi/bilibili-API-collect>
- 直播WS信息流：https://github.com/SocialSisterYi/bilibili-API-collect/blob/master/live/message_stream.md
- 直播数据包解析：https://segmentfault.com/a/1190000017328813?utm_source=tag-west

- 部分CMD包分析: <https://github.com/czp3009/bilibili-api/tree/master/record/%E7%9B%B4%E6%92%AD%E5%BC%B9%E5%B9%95>
- Qt解压zlib: <https://blog.csdn.net/doujianyoutiao/article/details/106236207>
- 实时关注: https://github.com/Banqijane/Bilibili_Danmuji
- 点歌姬网易云音源: <https://github.com/Binaryify/NeteaseCloudMusicApi>
- 点歌姬QQ音乐源: <https://github.com/jsososo/QQMusicApi>
- 点歌姬咪咕音乐源: <https://github.com/jsososo/MiguMusicApi>
- 点歌姬酷狗音乐源: https://www.cnblogs.com/daxiangxm/archive/2019/10/25/kugou_music_api.html
- 酷狗音乐下载地址: <https://cloud.tencent.com/developer/article/1795030>
- Star图标: <https://www.behance.net/gallery/83133509/Logo-animations-collaboration>
- 服务端模块: <https://github.com/nikhilm/qhttpserver>
- 抖音API接口: <https://github.com/skmcj/dycast>
- 抖音a_bogus解析: <https://github.com/ShilongLee/Crawler>
- TikTok/抖音/视频号/快手/虎牙的通用解析: 可遇AI: <https://mwmbav7xpt.feishu.cn/docx/Cdlfdi7Vdo72skx7B8FcM2Twnmb>