



RuleGo Editor 用户使用手册

规则链可视化编排 · 从搭建到调试的完整操作指南

V0.36

2026 年 9 月

RuleGo Team

目录

01 软件概述	1
1.1 产品简介	
1.2 运行环境	
1.3 登录与退出	
02 界面总览	7
2.1 工作区布局	
2.2 深浅主题	
2.3 「…」更多操作菜单	
2.4 用户菜单	
2.5 面板与布局	
03 创建与管理规则链	11
3.1 新建规则链	
3.2 资源管理	
3.3 规则链信息与分类	
3.4 规则链管理	
3.5 标签页操作	
3.6 导入与导出	
04 画布编排	19
4.1 组件库	
4.2 拖入节点	
4.3 连线	
4.4 选择、移动与对齐	
4.5 右键菜单	
4.6 缩放与视图	
05 节点配置	26
5.1 打开配置面板	
5.2 编写过滤脚本	
5.3 普通表单节点	
5.4 连线配置	
5.5 信息速览	
5.6 调试模式	
5.7 节点日志、文档与诊断	

06 保存与部署	34
6.1 脏态提示	
6.2 保存	
6.3 运行状态徽标	
6.4 下线与重新部署	
07 调试运行	38
7.1 调试面板	
7.2 发送与查看结果	
7.3 调试控制台	
7.4 未运行链的调试守卫	
08 运行记录	44
8.1 记录列表	
8.2 执行详情	
09 触发器与集成	46
9.1 触发方式怎么选	
9.2 输入端点组件	
9.3 定时触发	
9.4 内置 REST API 触发	
9.5 OpenAI 兼容接口	
9.6 MCP 集成	
10 组件、子规则链与共享节点	52
10.1 组件管理	
10.2 子规则链：把链当组件复用	
10.3 共享节点	
11 规则链市场	55
11.1 浏览市场	
11.2 导入模板	
12 AI 助手	57
12.1 打开聊天面板	
12.2 AI 生成规则链	
12.3 AI 助手设置	
12.4 全局技能	

13 效率工具与设置

60

- 13.1 快速切换规则链 (Ctrl+P)
- 13.2 跨链全局搜索 (Ctrl+Shift+F)
- 13.3 命令面板 (Ctrl+Shift+P)
- 13.4 快捷键一览
- 13.5 系统设置
- 13.6 关于

14 附录：服务端部署与配置

66

- 14.1 下载与启动
- 14.2 配置文件
- 14.3 账号与鉴权
- 14.4 组件市场
- 14.5 运行日志

01 软件概述

1.1 产品简介

RuleGo Editor 是 RuleGo 轻量规则引擎配套的**规则链可视化编排工具**。它把规则引擎的 JSON DSL（领域特定语言）变成一块可拖拽、可调试、可部署的画布：工程师无需手写配置文件，在浏览器里即可完成规则链的搭建、调试与上线。

主要能力：

- **可视化编排**：从组件库拖拽 150+ 内置组件（过滤、转换、动作、外部连接、AI、IoT 等），在画布上连线组成数据处理流程。
- **在线调试**：向规则链发送测试消息，逐节点查看消息流入流出，实时定位问题。
- **保存即部署**：保存后的规则链立即在服务端生效，支持上线、下线、重启。
- **运行记录**：每次执行的来源、耗时、成败与节点级明细自动留痕，可随时回看。
- **触发与集成**：定时、MQTT、Kafka 等端点组件接入，内置 REST / OpenAI 兼容 / MCP 接口供外部系统调用（第 09 章）。
- **AI 助手**：自然语言生成规则链、答疑与配置建议（第 12 章）。
- **规则链市场**：一键导入官方模板链，快速获得典型场景的参考实现。
- **多链管理**：资源树、多标签页、跨链全局搜索、快捷键与命令面板，适合长时间编排工作。

本手册以一位冷链物流物联网工程师的视角展开：从零搭建「冷链仓库温度监控」规则链——采集冷库温度遥测，温度超过 8℃ 自动记录高温告警，正常温度记录巡检日志——并在此过程中完整演示各项功能。

1.2 运行环境

项目	要求
客户端	Chrome / Edge 等现代桌面浏览器，分辨率建议 1440×900 及以上
服务端	RuleGo Server（编辑器与其同源部署，由服务端一并托管）
账号	由管理员分配（默认管理员账号 <code>admin</code> ）；服务端关闭鉴权时可直接进入

在浏览器地址栏打开部署地址（例如 `http://<服务器地址>:9090/editor/`）即可使用。服务端尚未部署？按第 14 章从下载到启动最快五分钟完成。

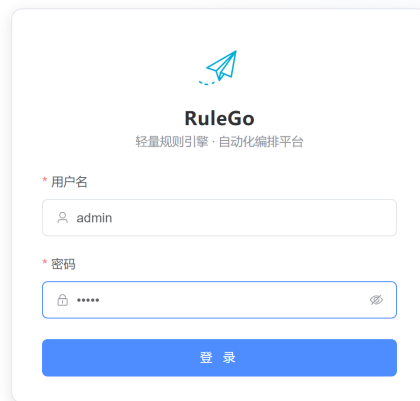
1.3 登录与退出

打开地址后，若服务端开启了鉴权，会先进入登录页：



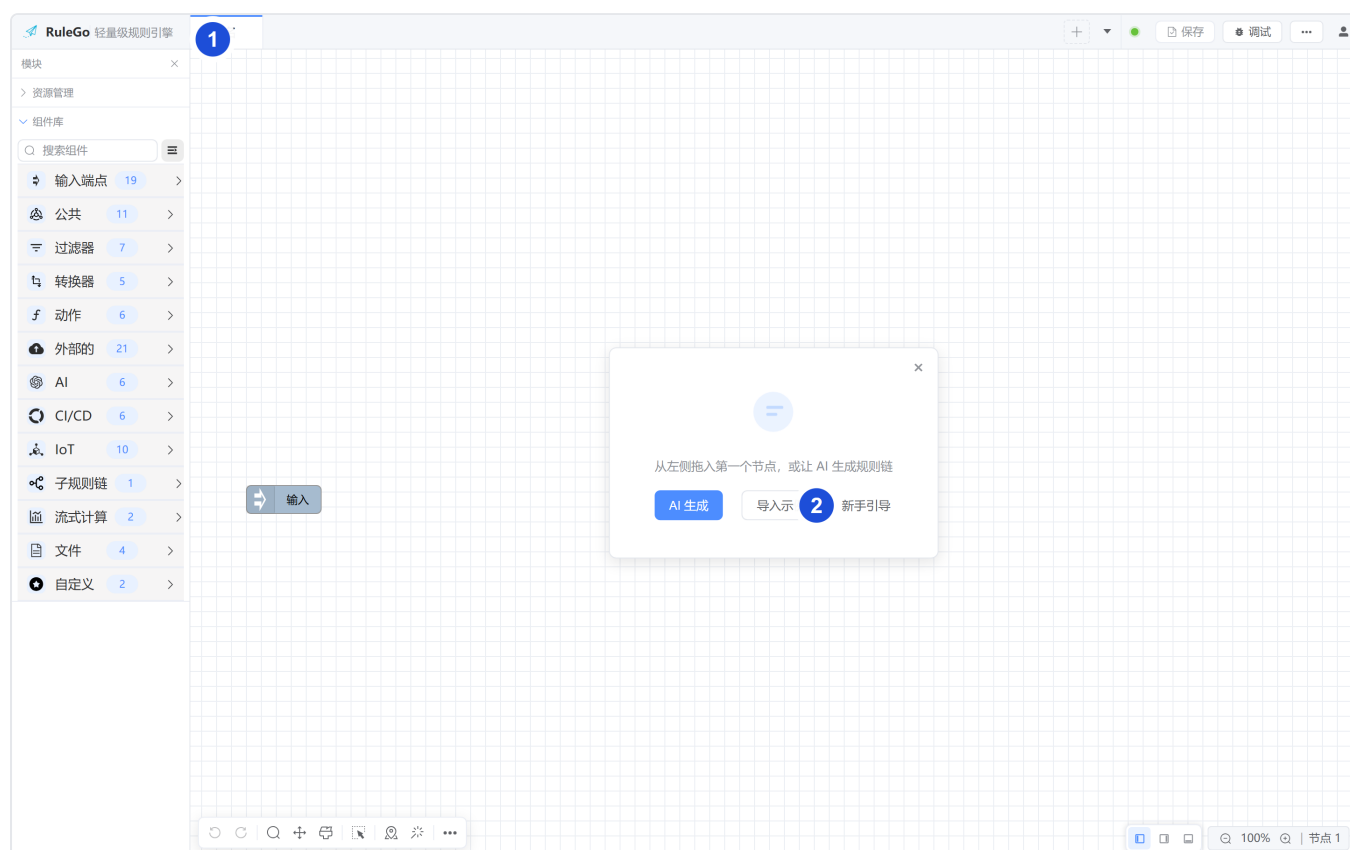
The image shows a login page for RuleGo. At the top, there is a logo consisting of a blue paper airplane icon and the text "RuleGo" in bold, with the tagline "轻量规则引擎 · 自动化编排平台" (Lightweight Rule Engine · Automated Orchestration Platform) below it. The page contains three main elements: a username input field with a red asterisk and the label "用户名" (Username), a password input field with a red asterisk and the label "密码" (Password), and a blue "登录" (Login) button. Each of these elements is preceded by a blue circular number indicating a step: "1" for the username field, "2" for the password field, and "3" for the login button. The input fields have placeholder text: "请输入用户名" (Please enter username) and "请输入密码" (Please enter password).

按提示输入账号信息，点击「登 录」：

The image shows a login form for RuleGo. At the top is the RuleGo logo, a stylized blue paper airplane, followed by the text "RuleGo" and "轻量规则引擎·自动化编排平台". Below this are two input fields. The first is labeled "用户名" (Username) with a red asterisk, and contains the text "admin". The second is labeled "密码" (Password) with a red asterisk, contains six dots, and has an eye icon on the right. At the bottom is a blue button with the text "登 录" (Login).

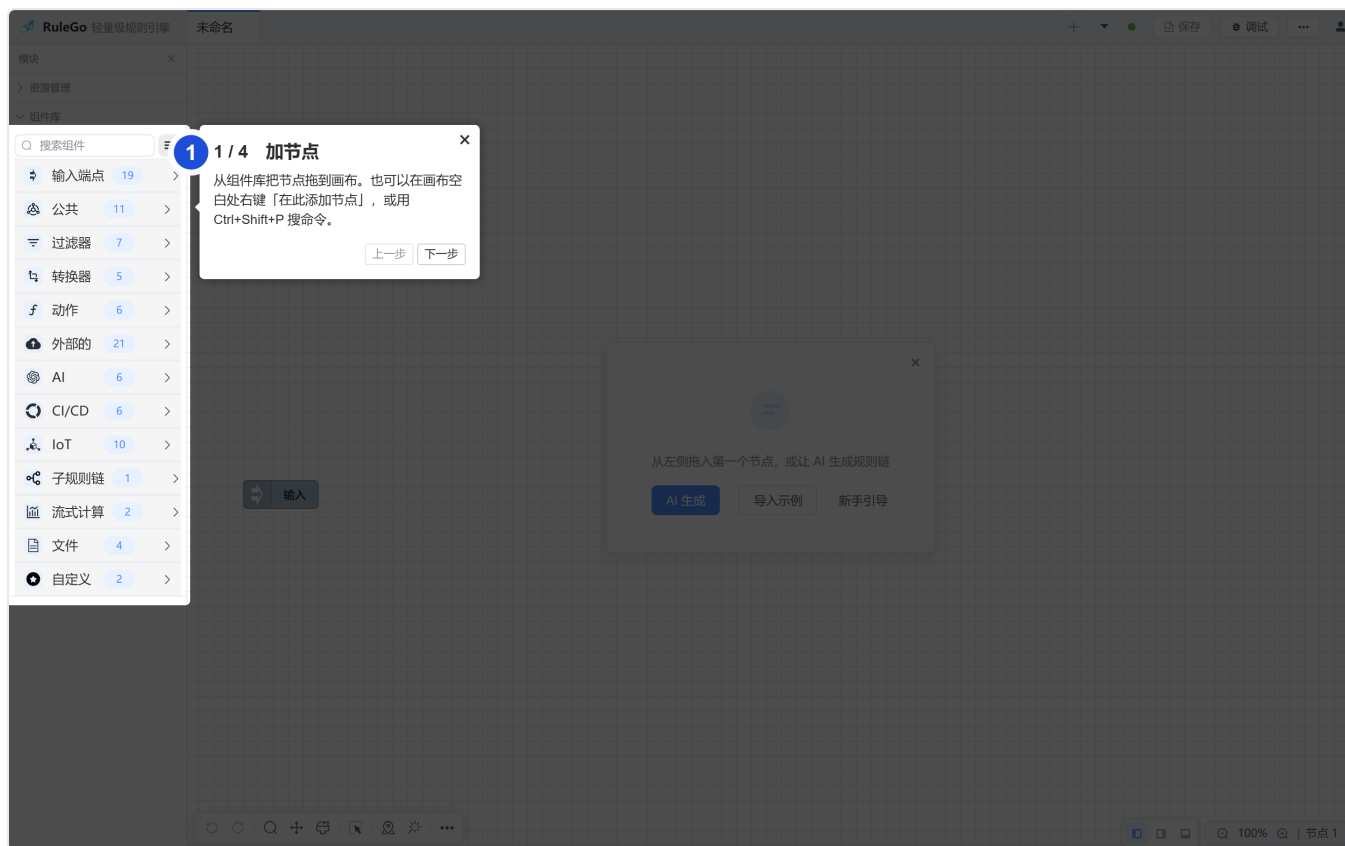
- ① **用户名**：输入管理员分配的账号。
- ② **密码**：输入对应密码。
- ③ **登 录**：点击后通过校验进入工作区；账号或密码错误会在表单下方提示。

登录成功后进入工作区。首次进入时系统自动开出一个「未命名」草稿标签页，画布中央提供「AI 生成」「导入示例」「新手引导」三个快速上手入口：



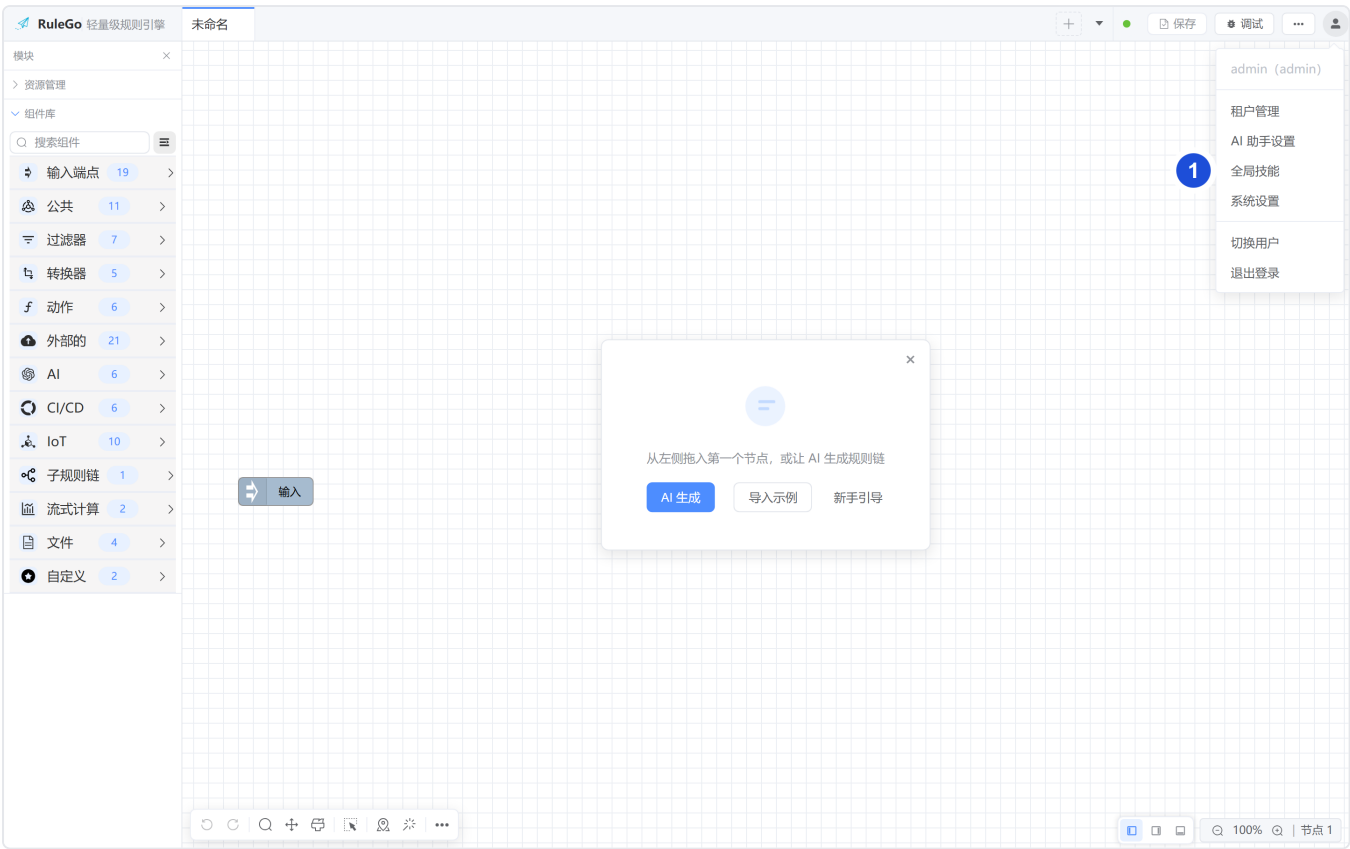
- ① **草稿标签页**：可以先在草稿上画节点，保存时再补名称与 ID。
- ② **新手引导**：首次使用建议点击，跟随 4 步引导快速认识画布、组件库与调试面板。

新手引导的第一步如下，点击「下一步」逐页了解，可随时通过气泡右上角的「×」退出：



- ① **引导气泡**：高亮当前讲解的界面区域，讲完自动进入下一步。

退出登录：点击顶栏右侧的用户胶囊，在弹出菜单中选择**退出登录**：



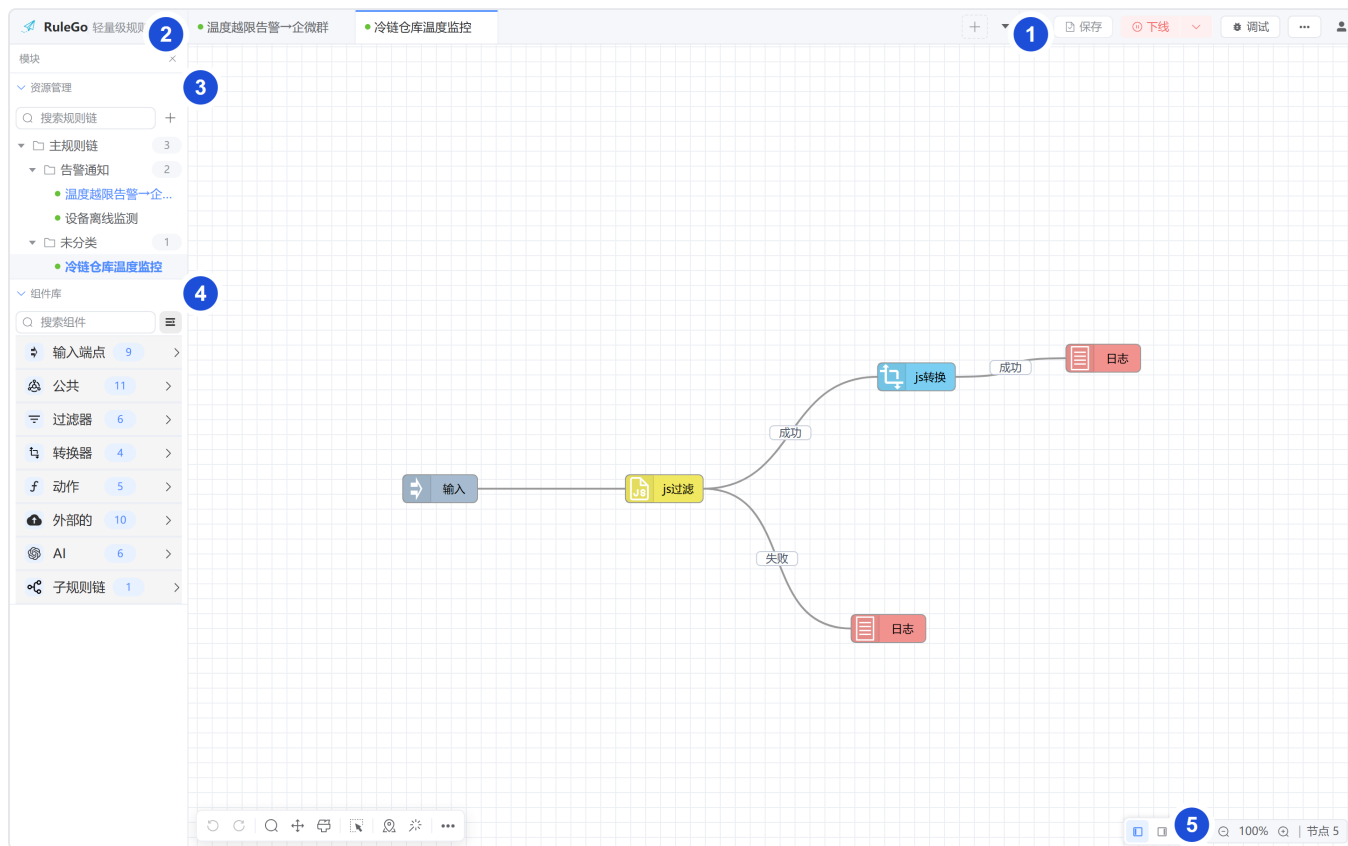
- ① **用户菜单**：包含租户管理、AI 助手设置、全局技能、系统设置、切换用户与退出登录。

提示：服务端未开启鉴权时不会出现登录页，所有请求以默认用户身份执行。

02 界面总览

2.1 工作区布局

打开一条规则链后，整个工作区如下图所示（浅色主题）：



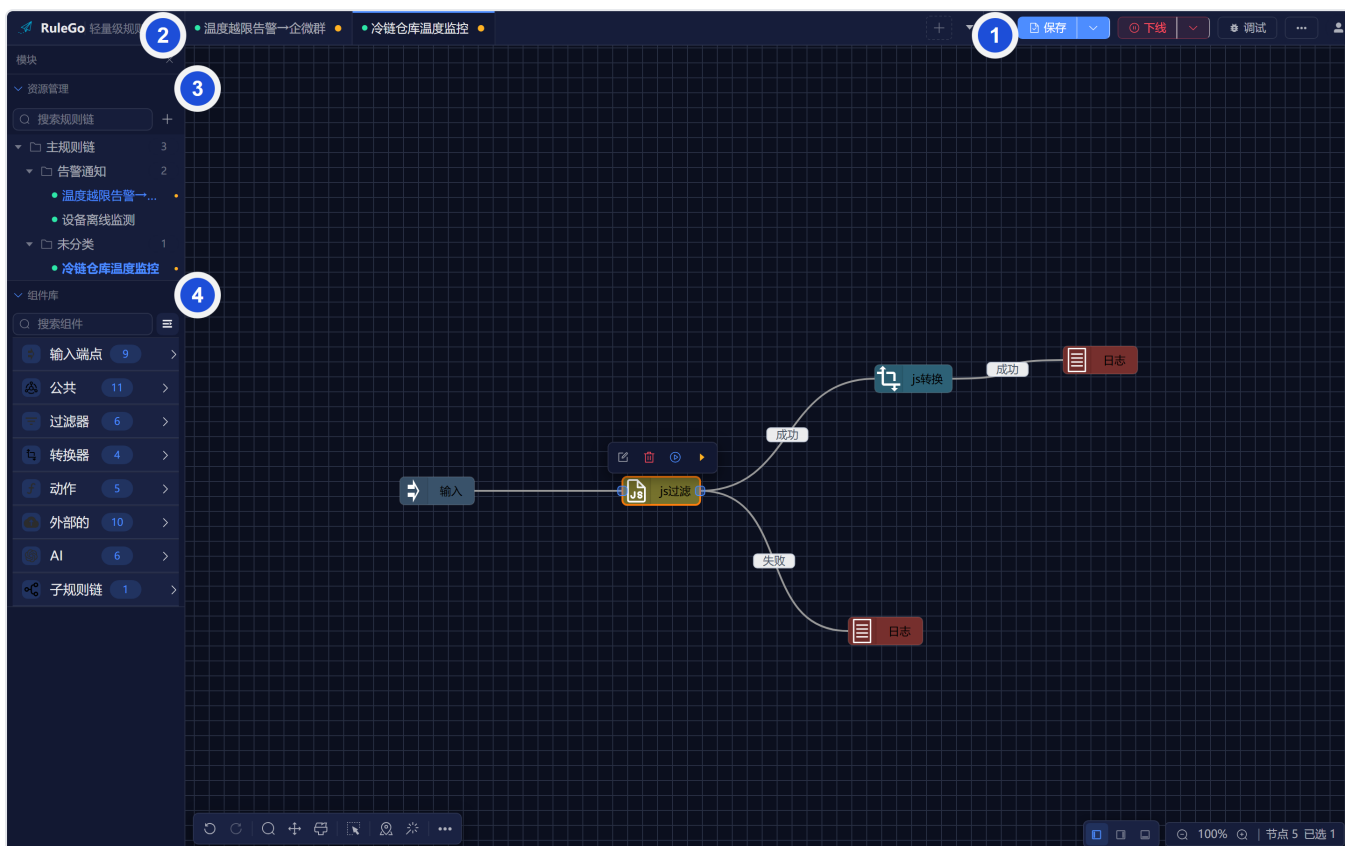
- ① **顶栏**：左侧是产品标识与「新建规则链」入口，右侧依次为连接状态灯、「保存」、「部署/下线」运行开关、「调试」与「…」更多操作菜单、用户胶囊。
- ② **标签页条**：每条打开的规则链一个标签，标签上显示运行状态徽标与未保存红点；「+」新建规则链，「v」列出全部标签。
- ③ **资源管理**：规则链树（按分类分组，含子规则链与共享节点）。
- ④ **组件库**：按功能分组的组件面板，是画布节点的来源。
- ⑤ **状态栏**：位于画布右下角，提供缩小、缩放比例（点击复位 100%）、放大等缩放控制，并显示画布节点数。

画布占据中央主区域，节点与连线都在这里编排。右侧面板（信息/配置/调试/日志/文档/诊断）与底部面板（调试/运行记录）默认收起，需要时通过画布工具栏的面板开关按钮展开。

提示：左侧整列称为「模块」栏，顶部有「收起」按钮可整列折叠，给画布腾出全宽。

2.2 深浅主题

界面提供浅色、深色与跟随系统三种外观。切换方式：点击顶栏「…」更多操作菜单 → **外观与帮助** → **主题**，选择「深色」等选项：

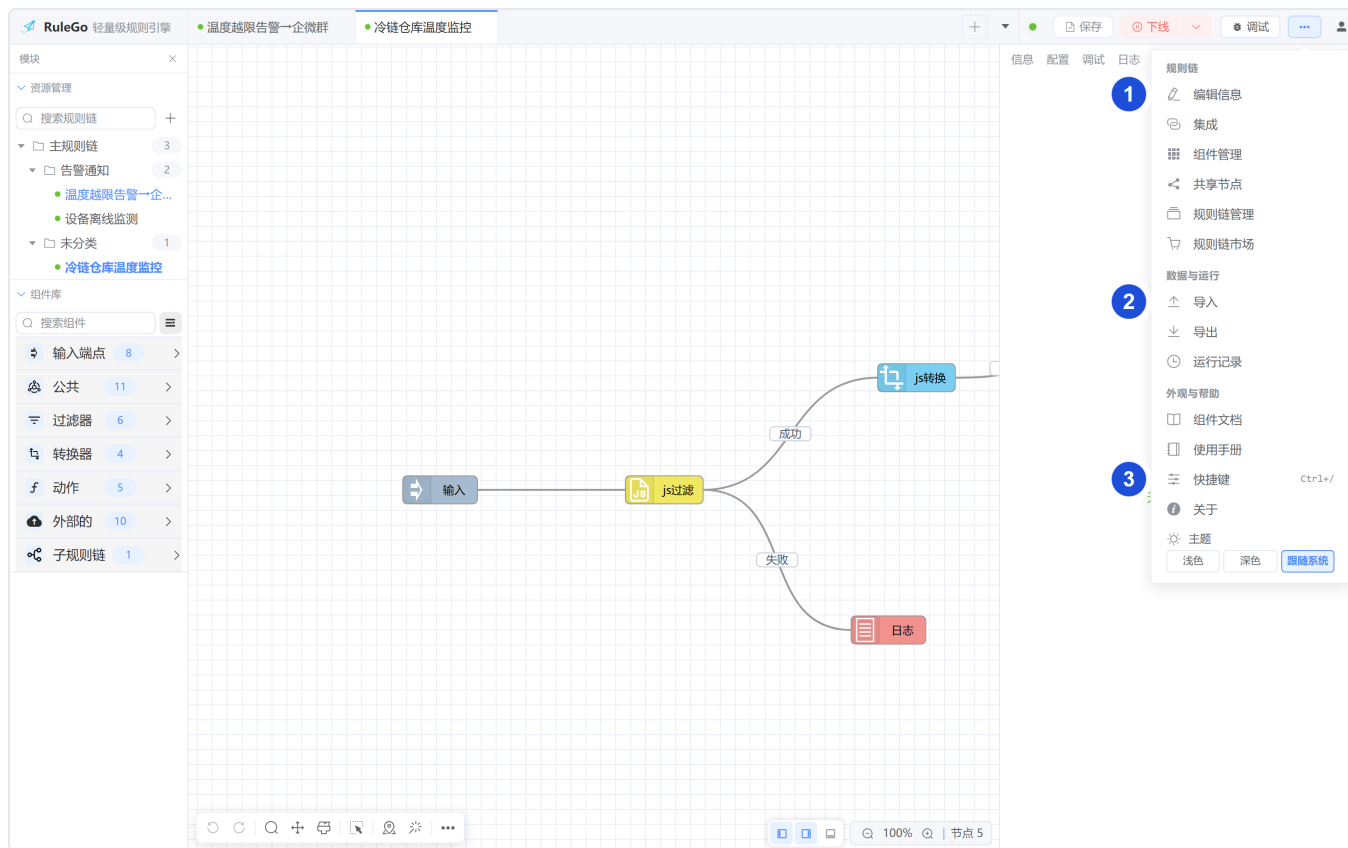


深色主题下所有面板、弹窗与画布同步换肤，功能布局完全一致（① 保存、② 标签页条、③ 资源管理、④ 组件库与浅色一一对应）。

提示：主题选择会自动记住，下次打开保持不变。

2.3 「…」更多操作菜单

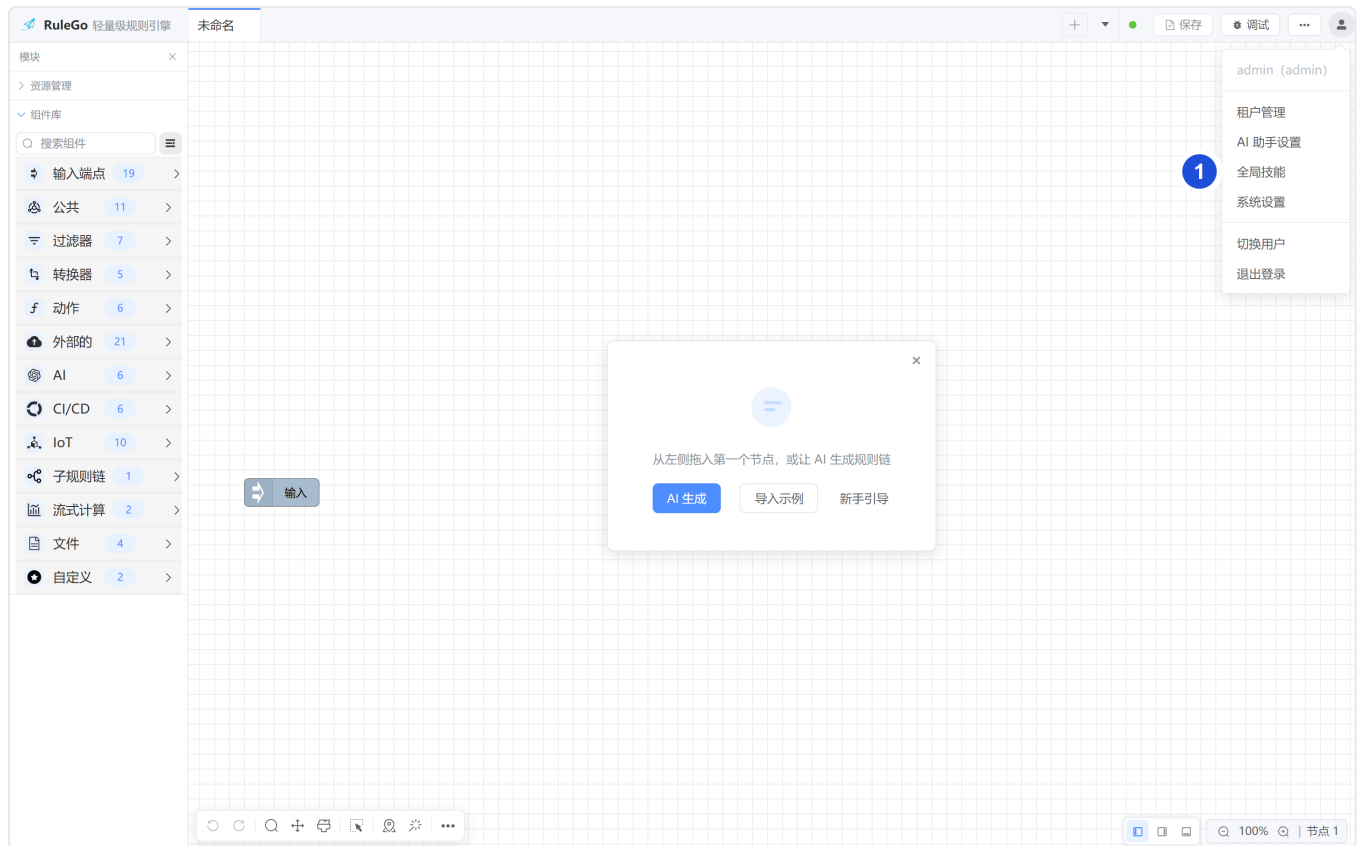
顶栏「…」按钮汇聚了全局功能，按三组组织：



- ① **规则链**：编辑信息（含分类与集成接口，见第 03、09 章）、组件管理、共享节点、规则链管理、规则链市场。
- ② **数据与运行**：导入 / 导出规则链 DSL、运行记录。
- ③ **外观与帮助**：组件文档、使用手册、快捷键（**Ctrl+ /**）、关于，组末尾是主题三态切换。

2.4 用户菜单

点击顶栏最右侧的用户胶囊：



- ① **用户菜单**：当前账号名（置灰显示）；**租户管理**（仅管理员，维护用户）；**AI 助手设置**（第 12 章）；**全局技能**（第 12 章）；**系统设置**（后端连接地址、语言等）；**切换用户 / 退出登录**。

2.5 面板与布局

编辑区由可开合的面板组成：左侧「资源管理 + 组件库」，右侧「信息 / 配置 / 调试 / 日志 / 文档 / 诊断」一组标签面板，底部「调试控制台 / 运行记录」。

开合方式有三处：画布左下工具栏的面板开关按钮、画布空白处**右键** → **面板子菜单**（逐个展开/收起左右与底部面板）、左侧「模块」栏顶部的「收起」按钮（整列折叠，给画布腾出全宽）。面板的开合状态会记住，下次打开保持上次布局。

03 创建与管理规则链

3.1 新建规则链

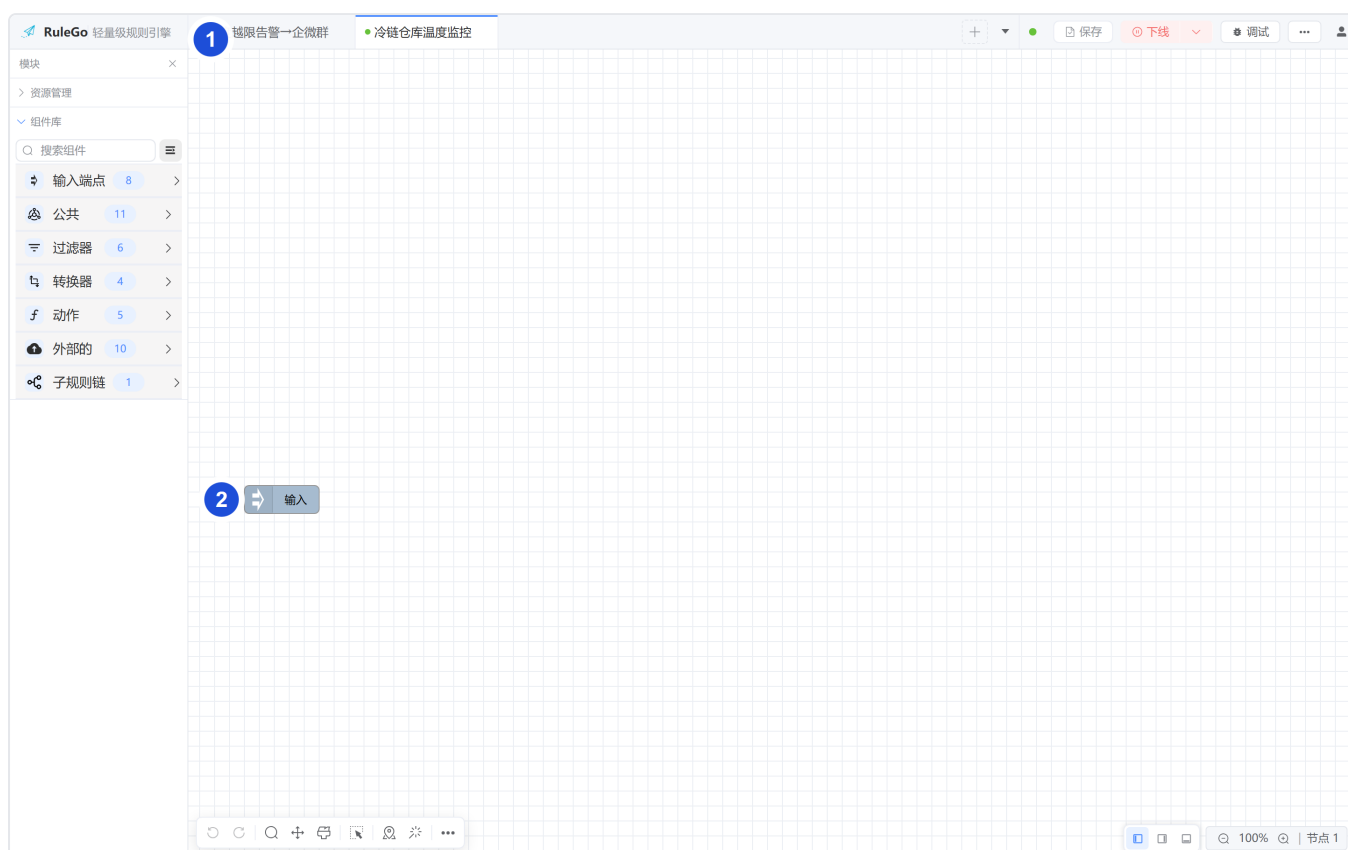
点击标签页条右侧的「+」按钮，弹出新建对话框（服务器上还没有任何规则链时，「+」会直接开出一个草稿标签页，先画后命名即可）：

填写规则链基本信息：



- ① **名称**：规则链的显示名，如「冷链仓库温度监控」。
- ② **ID**：规则链唯一标识，留空自动生成；自定义时只能使用字母、数字、下划线等安全字符。
- ③ **描述**：用途说明，会显示在链列表与资源树提示中。
- ④ **创建**：确认后规则链保存到服务端并**立即上线运行**。

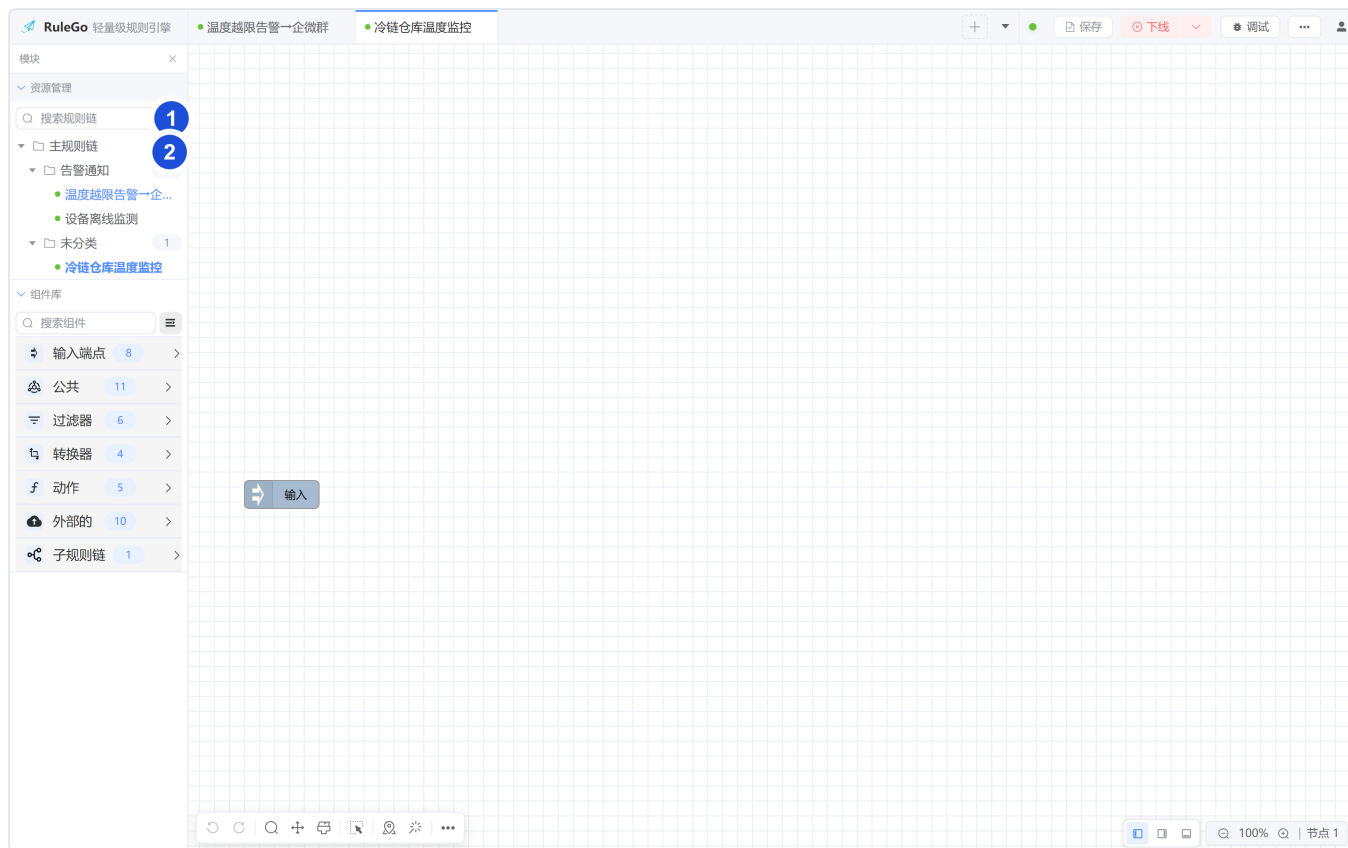
创建成功后，新链自动在标签页中打开，画布初始包含一个「输入」节点，代表规则链的消息入口：



- ① **新链标签页**：标题即规则链名称；顶栏出现「下线」按钮，说明链已在运行。
- ② **画布**：当前只有输入节点，从左侧组件库拖入下一个节点即可开始编排（见第 04 章）。

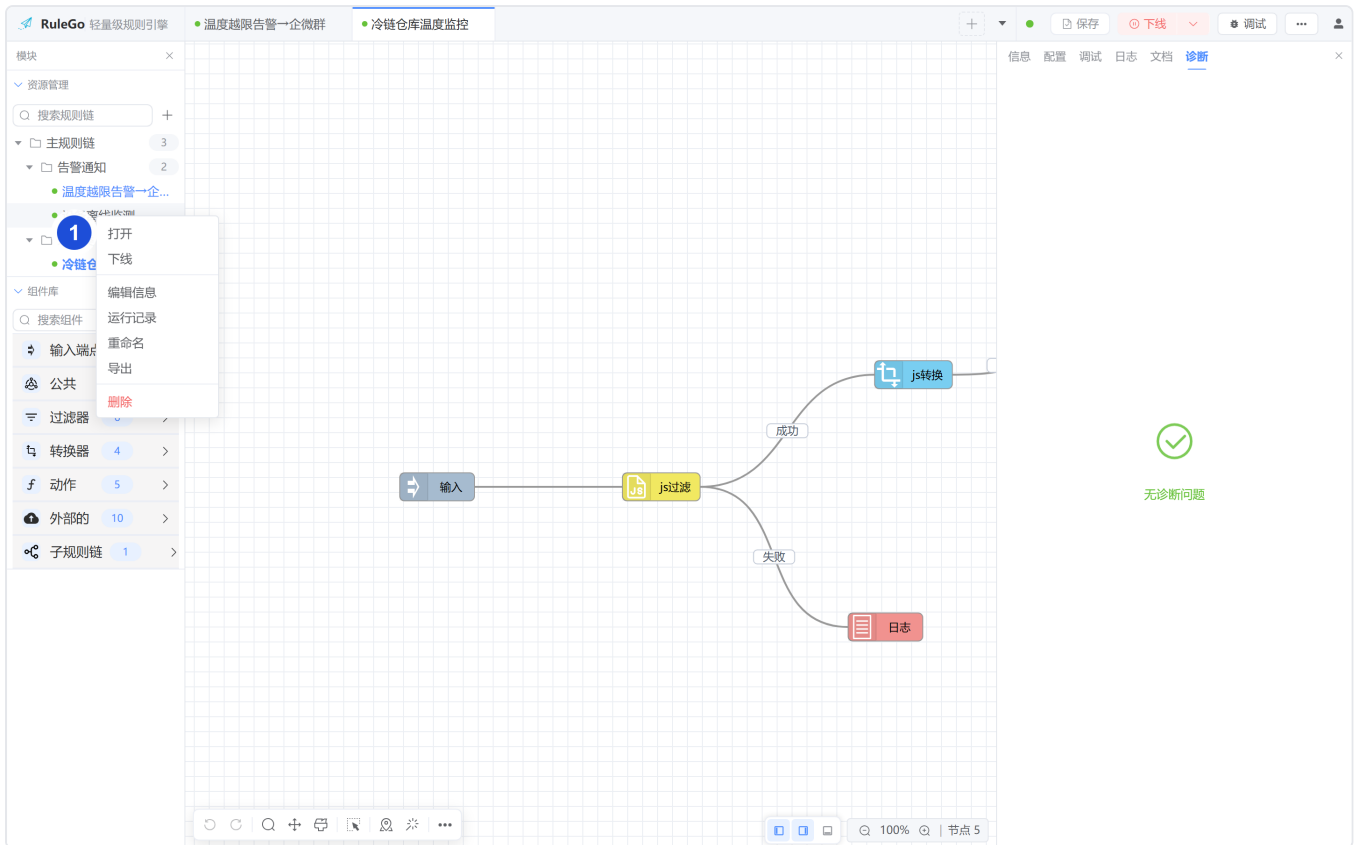
3.2 资源管理

点击左侧「资源管理」面板标题展开规则链树：



- ① **搜索框**：按名称或 ID 过滤规则链。
- ② **主规则链**：按分类分组（如「告警通知」「未分类」），括号内是链数量；点击链名即在标签页中打开。链名前的圆点是运行状态：绿色运行中、灰色已下线。

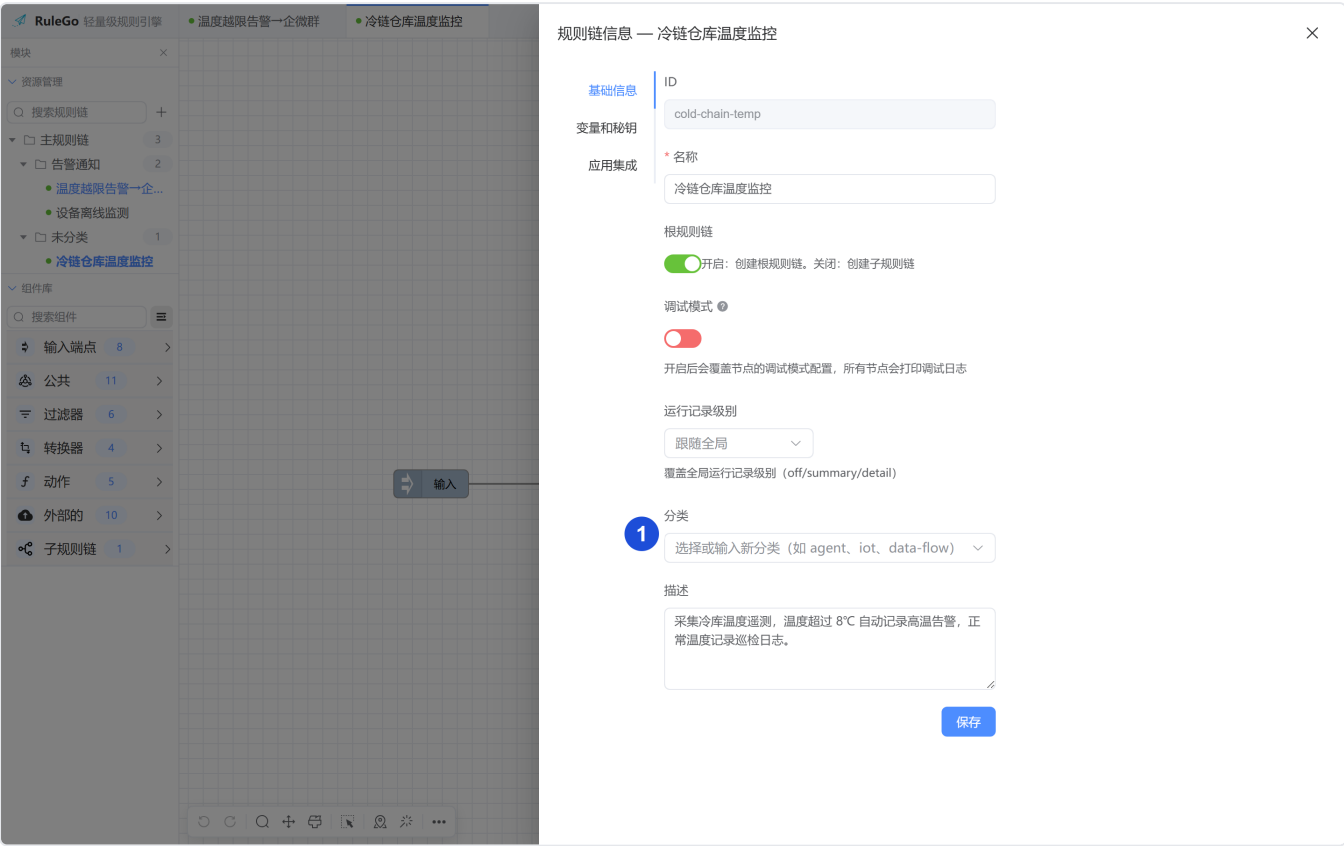
在树的链节点上**右键**，弹出快捷操作菜单：



- ① **快捷操作**：重命名、编辑信息、导出、查看运行记录、上线/下线、删除——多数日常运维不必打开链就能完成。

3.3 规则链信息与分类

链名、分类、描述等元信息在「规则链信息」抽屉维护。入口：资源树右键 → **编辑信息**，或顶栏「…」菜单 → **规则链** → **编辑信息**：



- ① **分类**：下拉可直接选已有分类，也可输入新分类名创建（如 **告警通知**）；资源树按分类分组展示，链多了以后靠分类保持秩序。

抽屉左侧还有两个标签：

- **变量和密钥**：维护链级变量（模板里以 `${global.变量名}` 引用）与密钥，敏感配置不必写死在节点参数里。
- **应用集成**：本链对外提供的接口清单（REST/OpenAI/MCP），见第 09 章。

3.4 规则链管理

批量查看与运维走「规则链管理」抽屉：顶栏「…」更多操作菜单 → 规则链 → 规则链管理：

模块

资源管理

搜索规则链

主规则链

告警通知

温度超限告警一企...

设备离线监测

未分类

冷链仓库温度监控

组件库

搜索组件

输入端点

公共

过滤器

转换器

动作

外部的

子规则链

输入

规则链管理

1 规则链

子规则链

新建规则链

状态

请输入关键字

ID	名称	分类	状态	更新时间	操作
cold-chain-temp	冷链仓库温度监控		运行	2026/09/20 10:59:24	设计 下线 删除
market-temp-alarm-wecom	温度超限告警一企微群	告警通知	运行	2026/09/20 10:34:04	设计 下线 删除
market-offline-detect	设备离线监测	告警通知	运行	2026/09/18 18:49:41	设计 下线 删除

共 3 条

20条/页

1

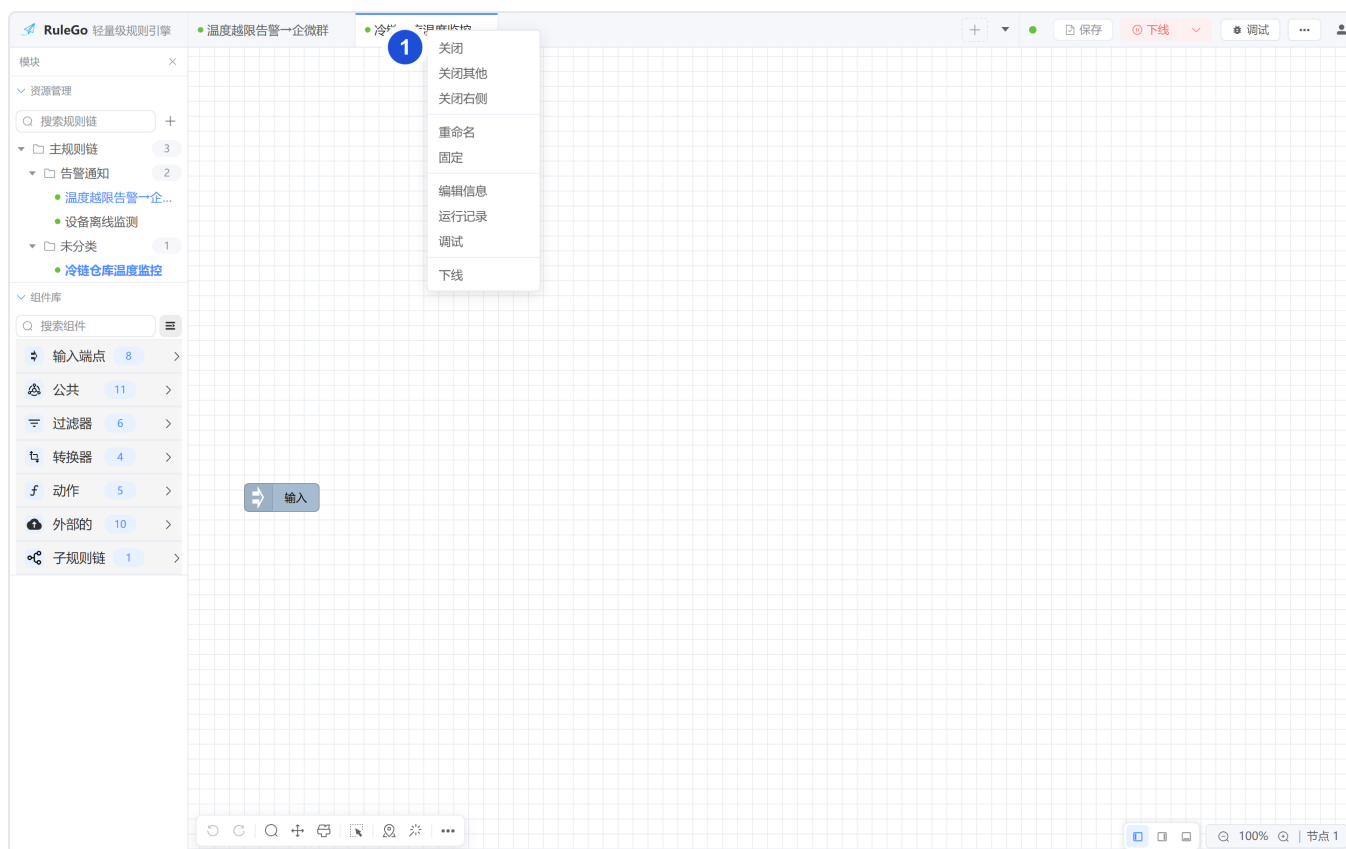
前往

1 页

- ① 「规则链 / 子规则链」标签：切换两类链的列表。
- ② 链列表：每行显示名称、ID、分类、更新时间与运行状态，右侧提供上线/下线、编辑信息、删除等操作；顶部支持按名称搜索、按状态筛选、新建规则链。

3.5 标签页操作

同时编辑多条链时，在标签页上**右键**可以批量管理：



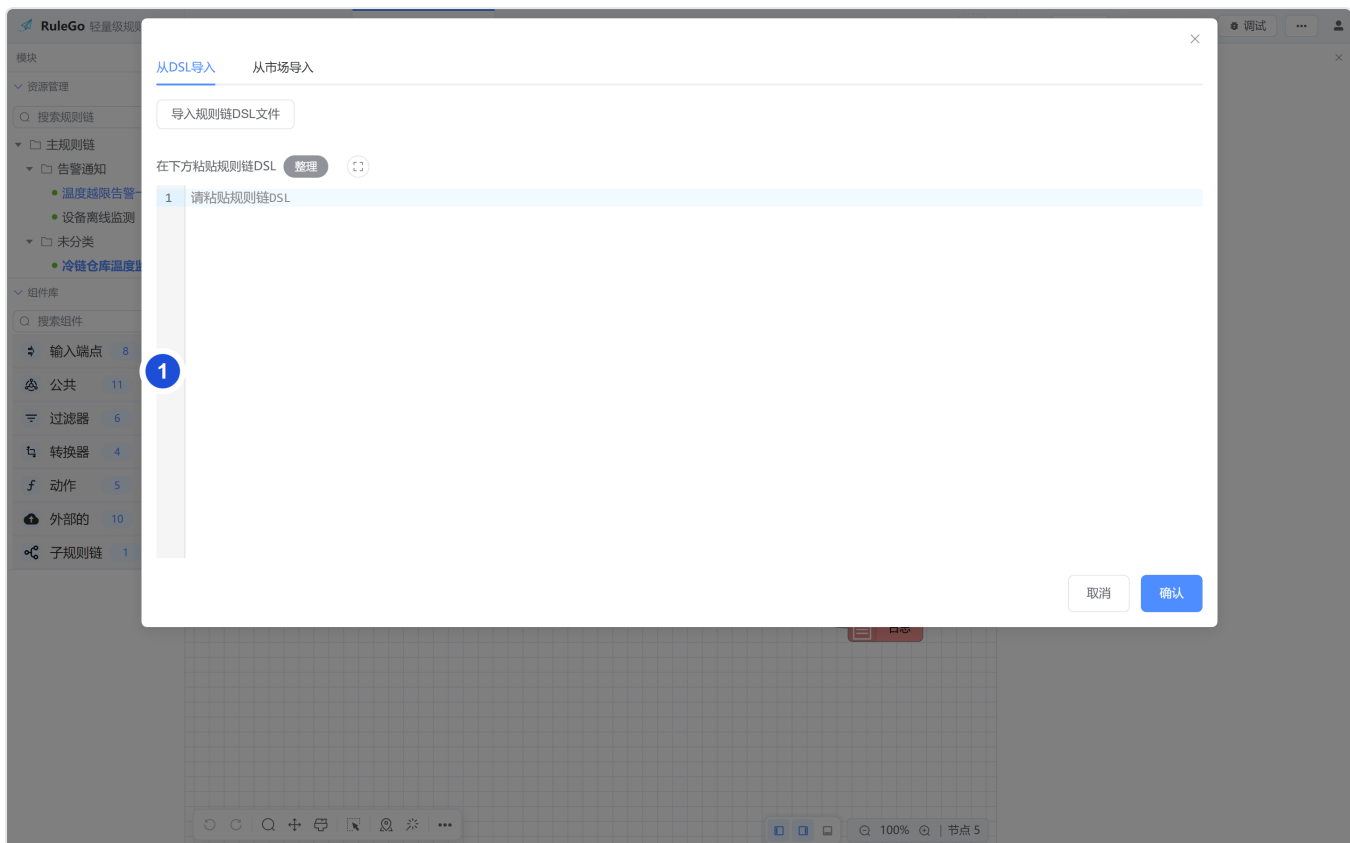
- ① **右键菜单**：关闭、关闭其他、关闭右侧、固定/取消固定、重命名、编辑链信息。

提示：标签页最多同时打开 8 条链，超出的按最近使用自动淘汰；被修改未保存的标签带红点提示（见第 06 章）。

3.6 导入与导出

链可以 DSL 文件的形式流转。入口：顶栏「…」更多操作菜单 → **数据与运行**：

- **导出**：把当前链保存为 `.json` DSL 文件，用于备份或迁移到其他环境。
- **导入**：选择 DSL 文件或直接粘贴 DSL 内容，确认后链即入库并打开：



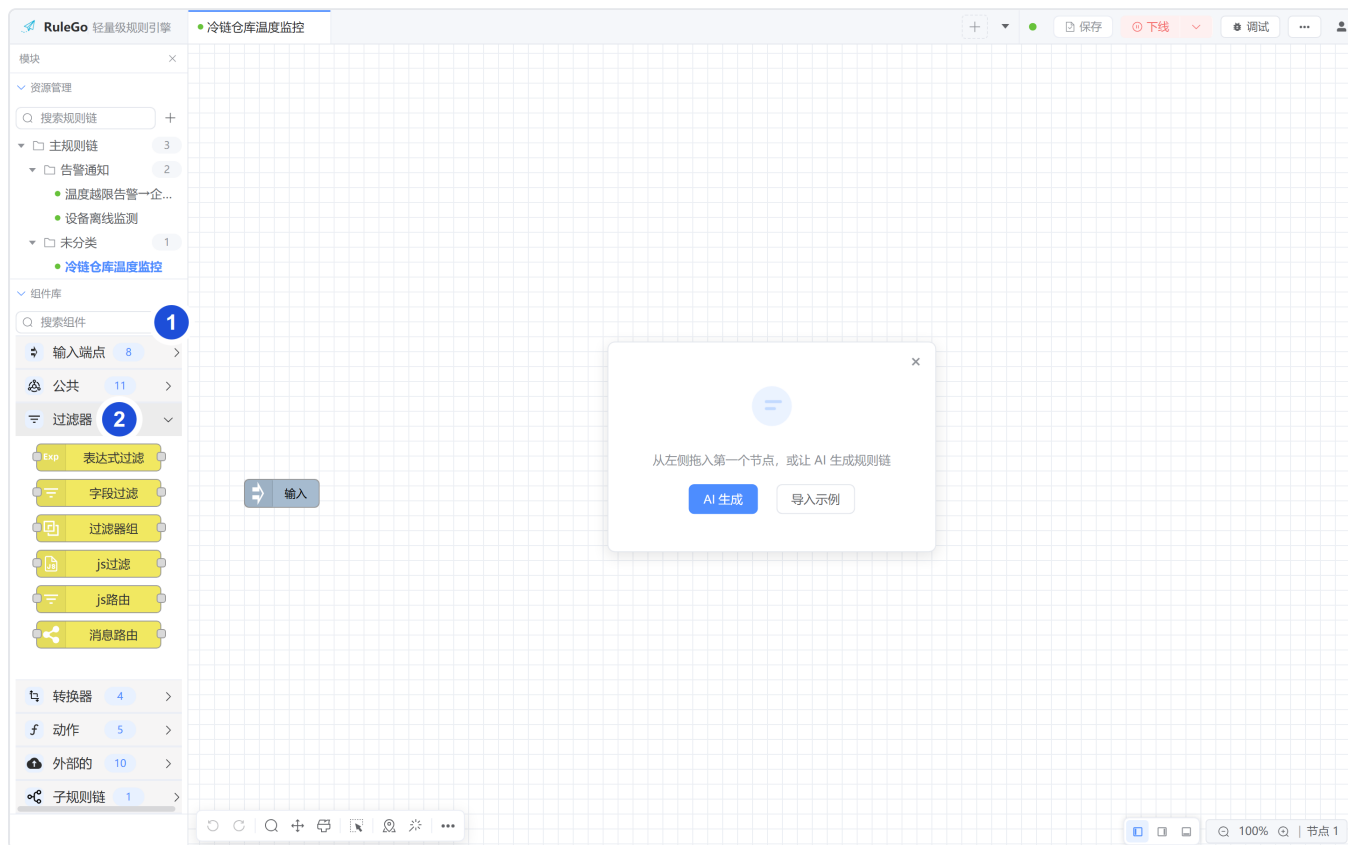
- ① **DSL 编辑区**：粘贴或自动载入的规则链 JSON，确认无误后导入。

提示：导入市场模板是同一种机制的「精选版」，见第 11 章。

04 画布编排

4.1 组件库

左侧「组件库」按功能分组列出了所有可用组件：输入端点、公共（流程控制）、过滤器、转换器、动作、外部的、AI、IoT 等。点击分组名展开，或在搜索框输入关键字快速定位：

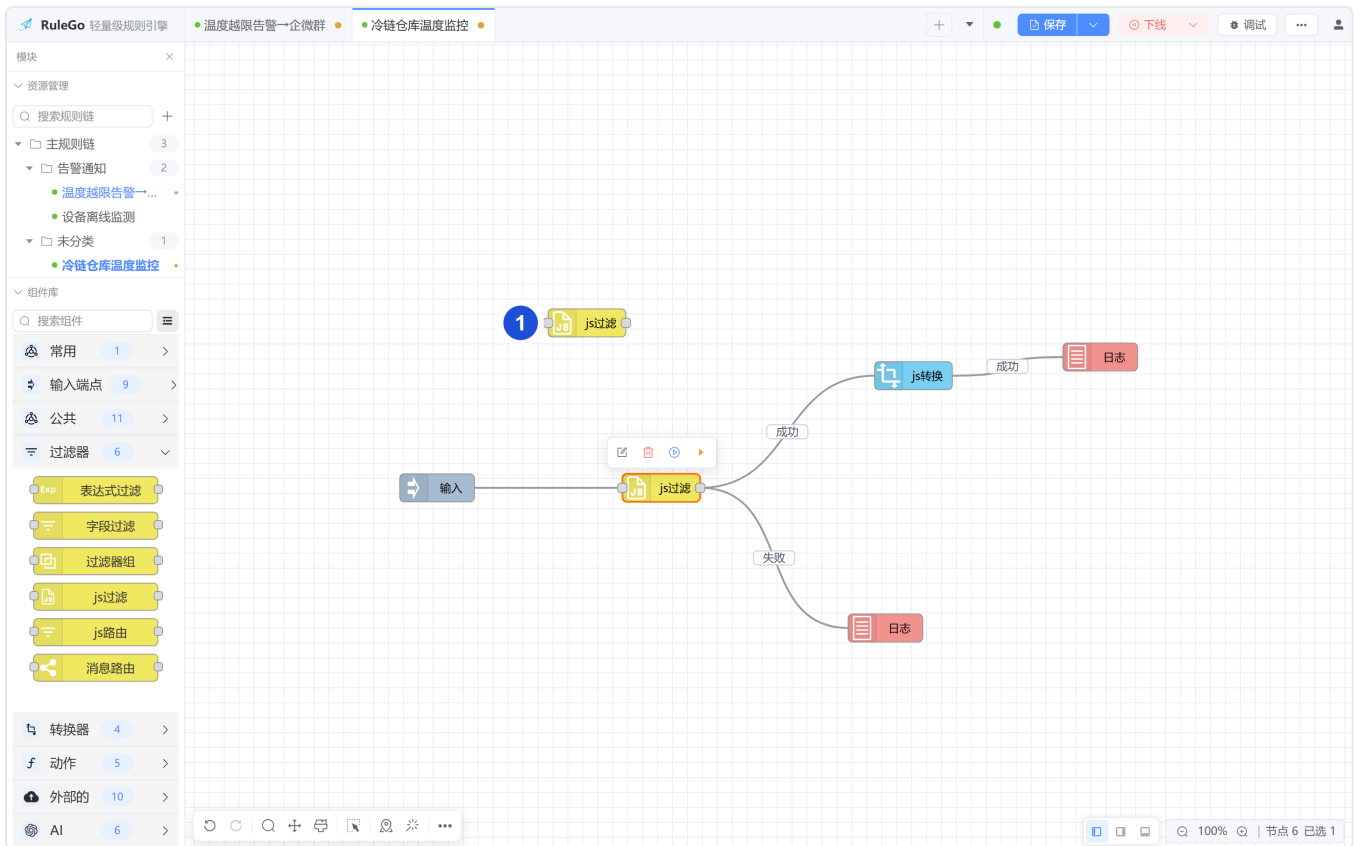


- ① **搜索框**：输入组件名（如「js过滤」）自动展开命中的分组。
- ② **分组**：如「过滤器」组包含 js 过滤、表达式过滤、字段过滤、js 路由等。

悬停组件卡片可查看组件说明。冷链监控链需要的三个组件都在常用分组里：**js过滤**（过滤器组）、**js转换**（转换器组）、**日志**（动作组）。

4.2 拖入节点

把组件卡片**按住拖到画布**目标位置松开，即添加一个节点。先拖入「js过滤」：

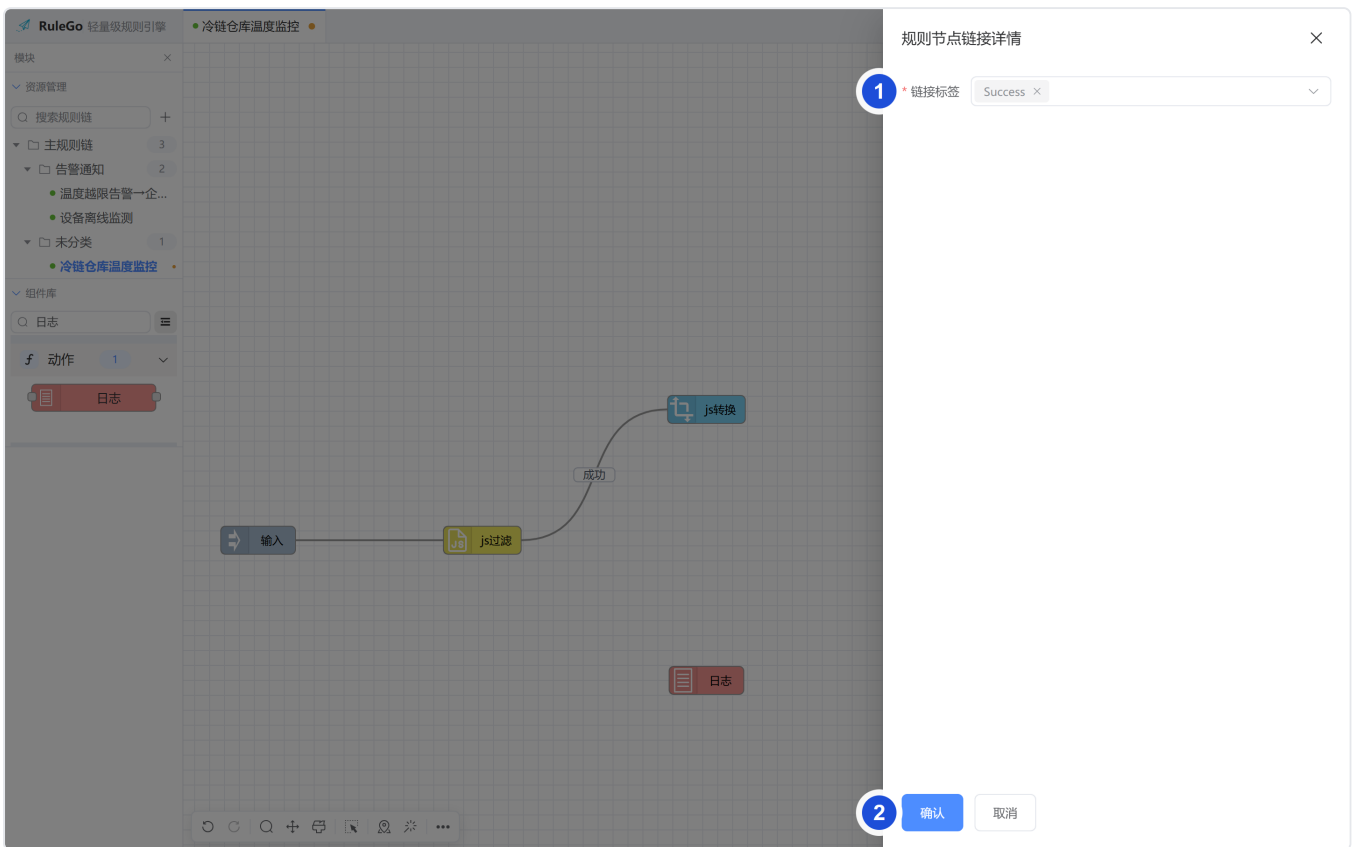


- ① **新节点**：落在松开鼠标的位置，节点标题显示组件名，默认自动命名（如「节点 2」）。

重复拖入：再拖一个「js转换」、两个「日志」节点。画布空白处右键选择「在此添加节点」也可以按类型快速加节点。

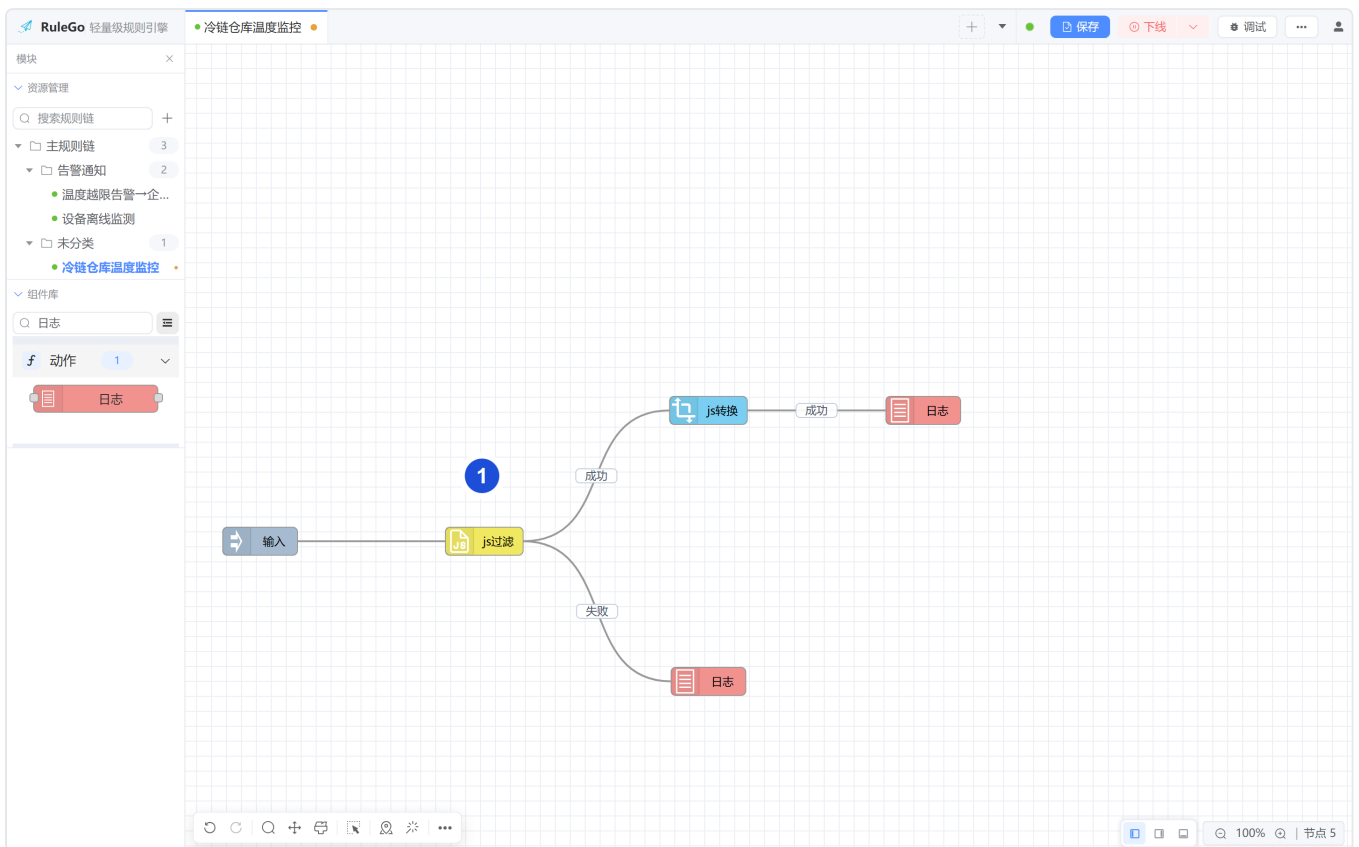
4.3 连线

把鼠标移到节点边缘的锚点，按住拖到目标节点松开即建立连线。连线后弹出「规则节点链接详情」确认标签：



- ① **链接标签**：选择这条连线承载的关系类型，如 js 过滤节点的「成功 / 失败」。
- ② **确认**：保存标签；连线中段会显示标签文字。

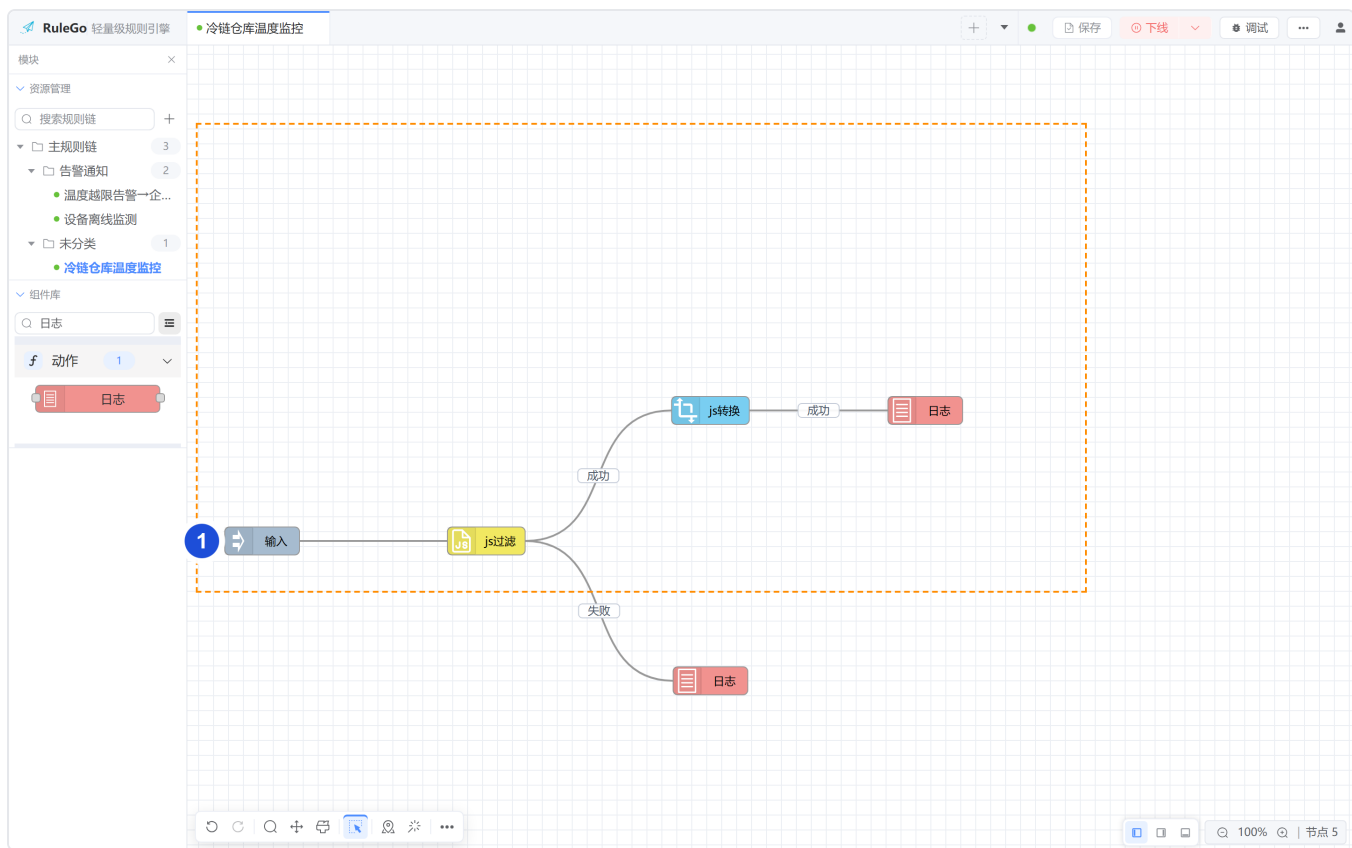
按冷链监控的逻辑连线：输入 → js过滤；js过滤「成功」→ js转换；js转换「成功」→ 日志（高温告警）；js过滤「失败」→ 日志（温度正常）。完成后画布如下：



- ① **连线**：中段文字即关系标签（成功/失败），消息按判定结果走对应分支。

4.4 选择、移动与对齐

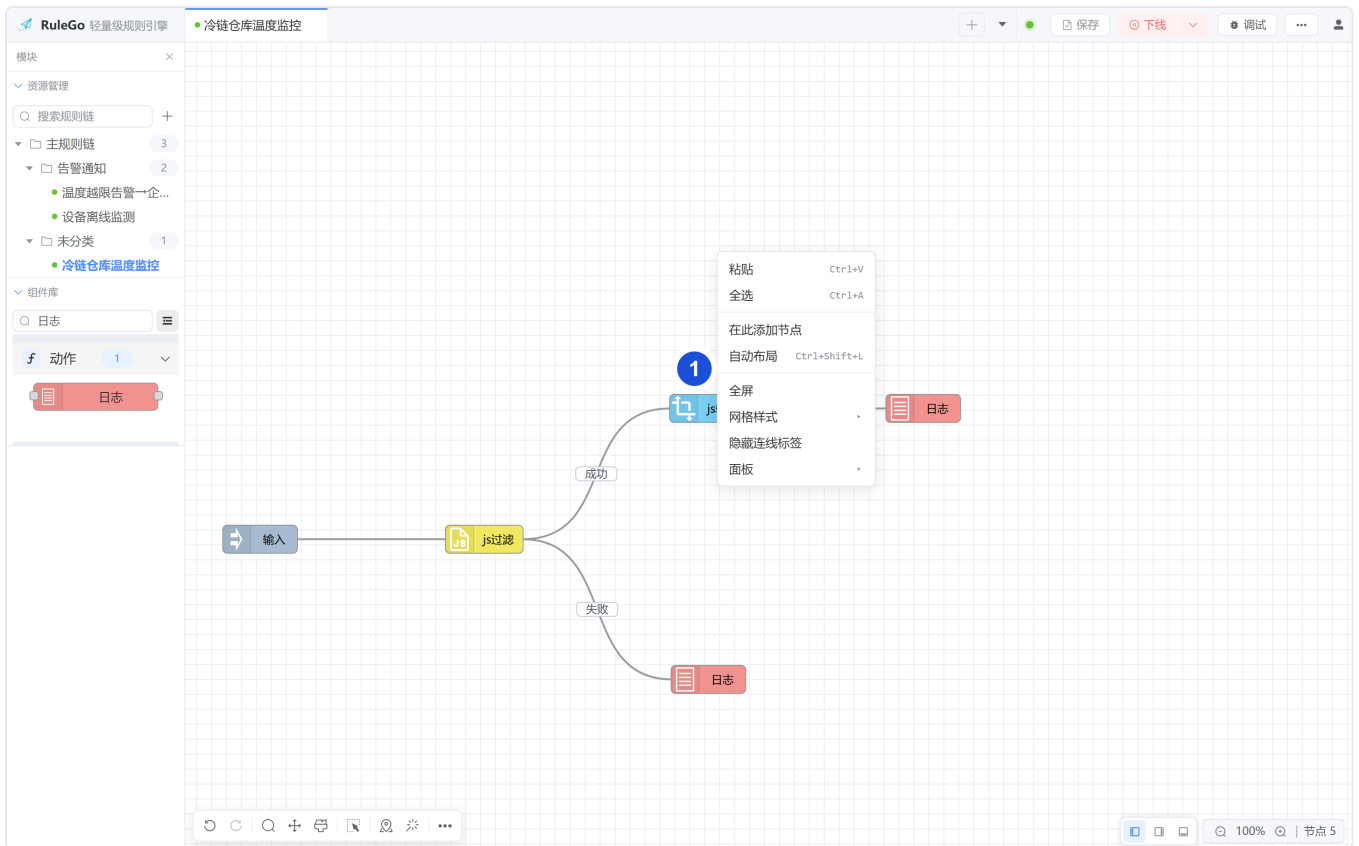
- **单击节点或连线选中**；**空白处按住拖动框选**多个节点：



- ① **框选**：虚线框内的节点一起选中，可整体拖动、删除或复制。
- 按住节点拖动调整位置，移动时显示对齐参考线；**Ctrl+C** / **Ctrl+V** 复制粘贴，**Ctrl+Z** 撤销，**Delete** 删除选中元素。

4.5 右键菜单

在画布空白处右键：

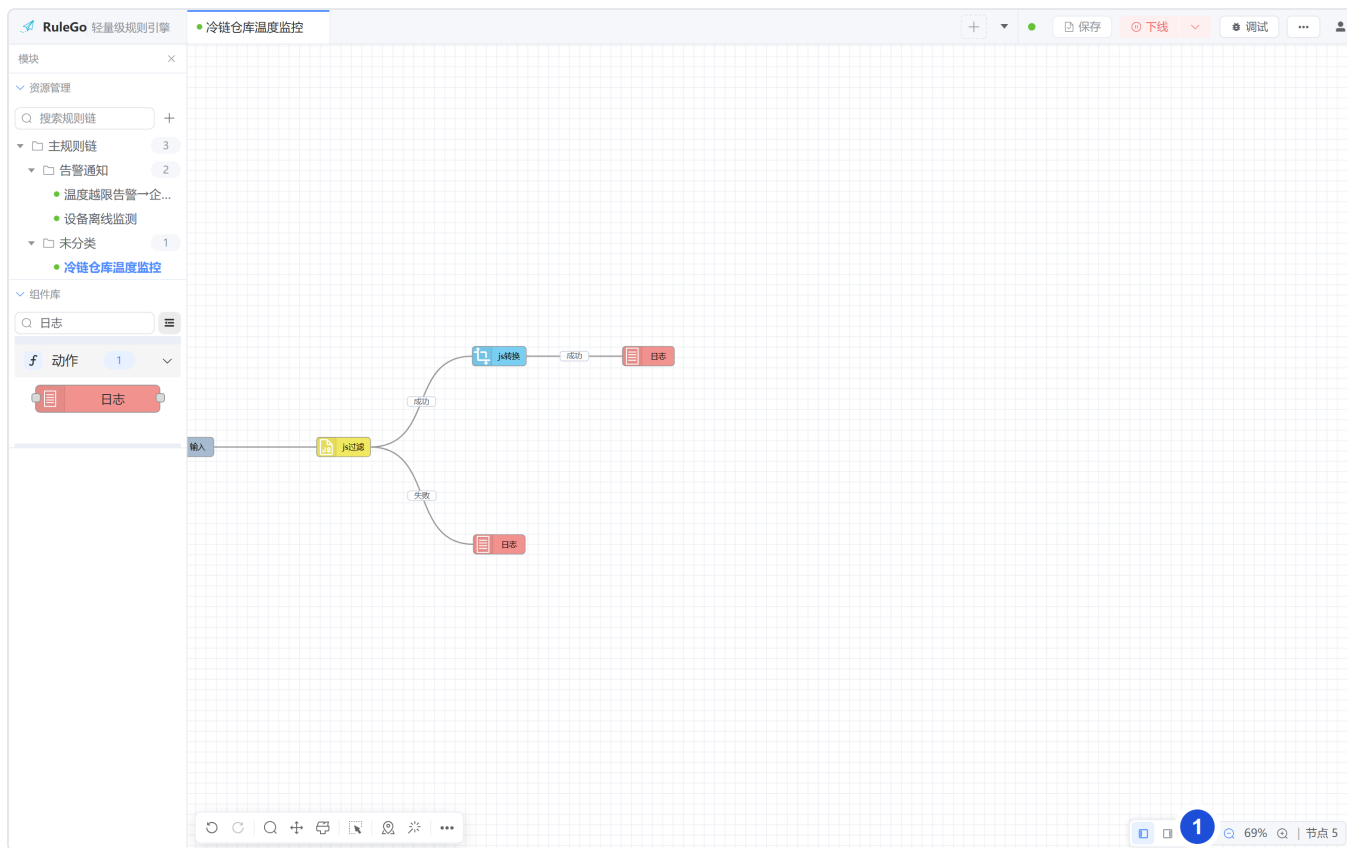


- ① **画布菜单**：包含在此添加节点、粘贴、全选、自动布局、全屏、网格样式、隐藏连线标签，以及「面板」子菜单（展开/收起左右与底部面板）。

节点与连线上的右键菜单则提供编辑、删除、从此节点运行等针对性操作。

4.6 缩放与视图

画布右下角状态栏控制视图：

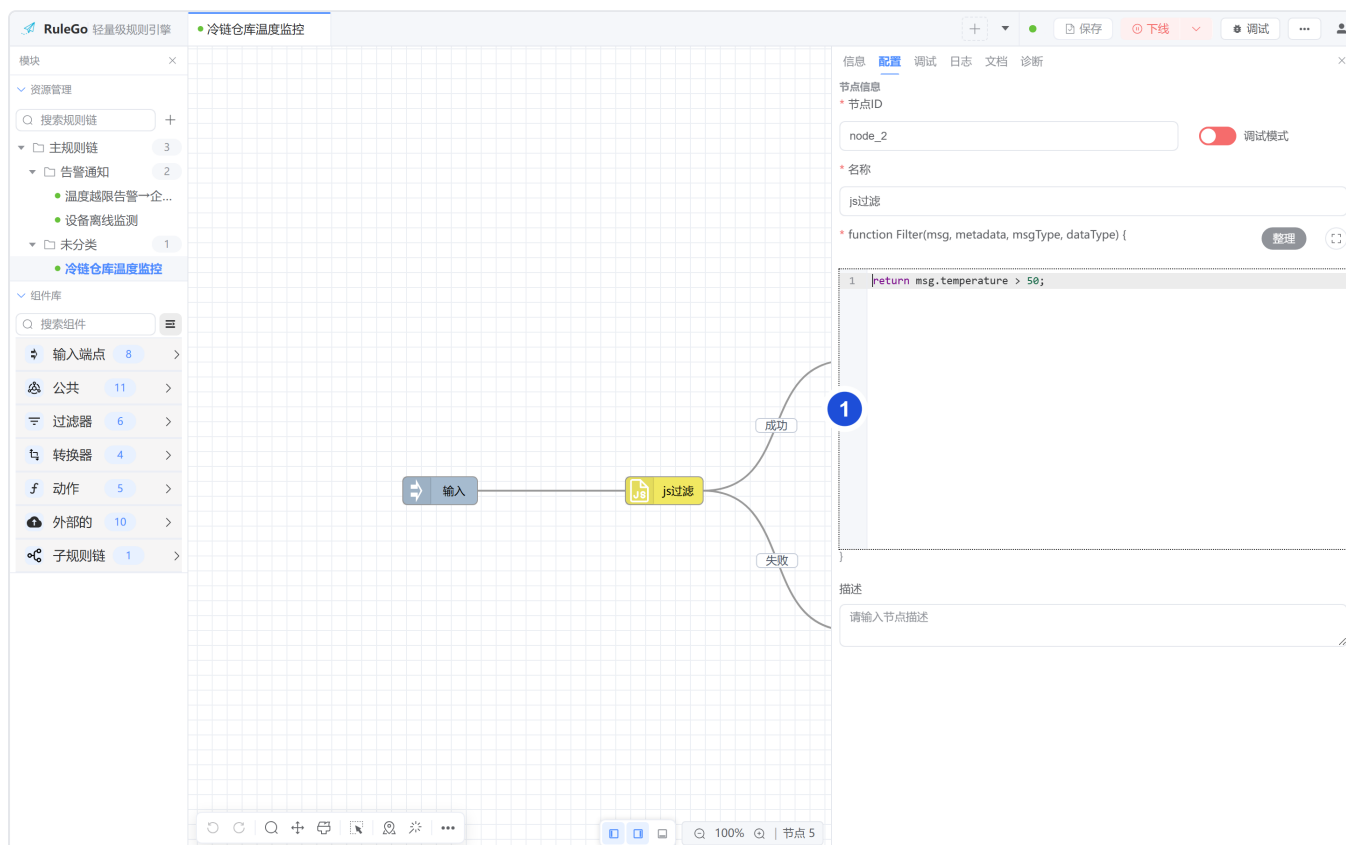


- ① **缩放控件**：「-」缩小、中间百分比点击回到 100%、「+」放大；快捷键 `Ctrl + +/-`，`Ctrl + 0` 复位。

05 节点配置

5.1 打开配置面板

双击节点，右侧面板自动展开并切到「配置」标签。双击「js过滤」节点：

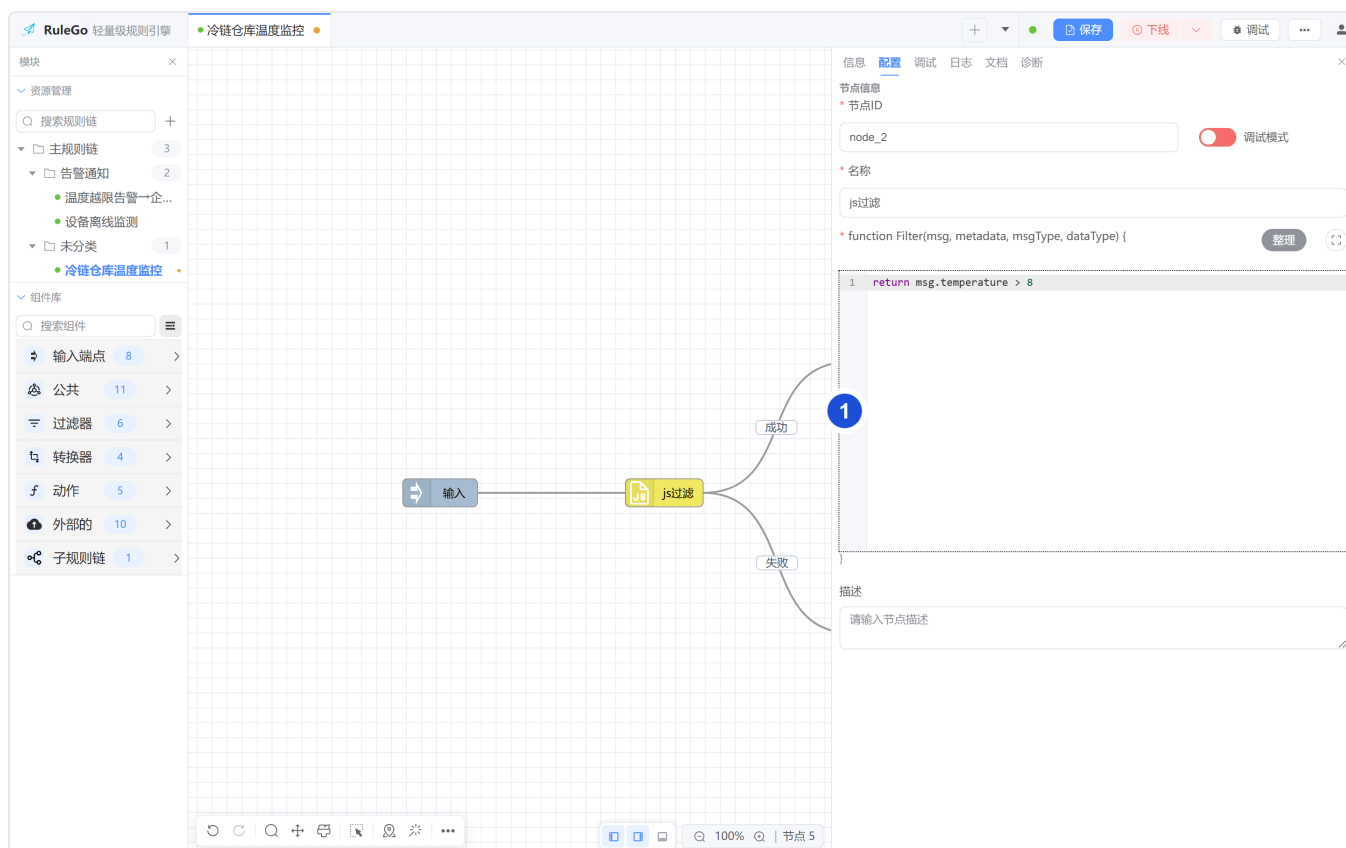


- ① **脚本编辑器**：js 过滤的核心配置，编写判定函数的函数体。

面板上方还有节点级公共字段：节点 ID、名称、调试模式开关。改动的字段会自动回传画布，节点上即刻生效。

5.2 编写过滤脚本

js 过滤节点的脚本返回 `true` 时消息走「成功」连线，否则走「失败」连线。把判定条件改为冷链安全上限 8℃：



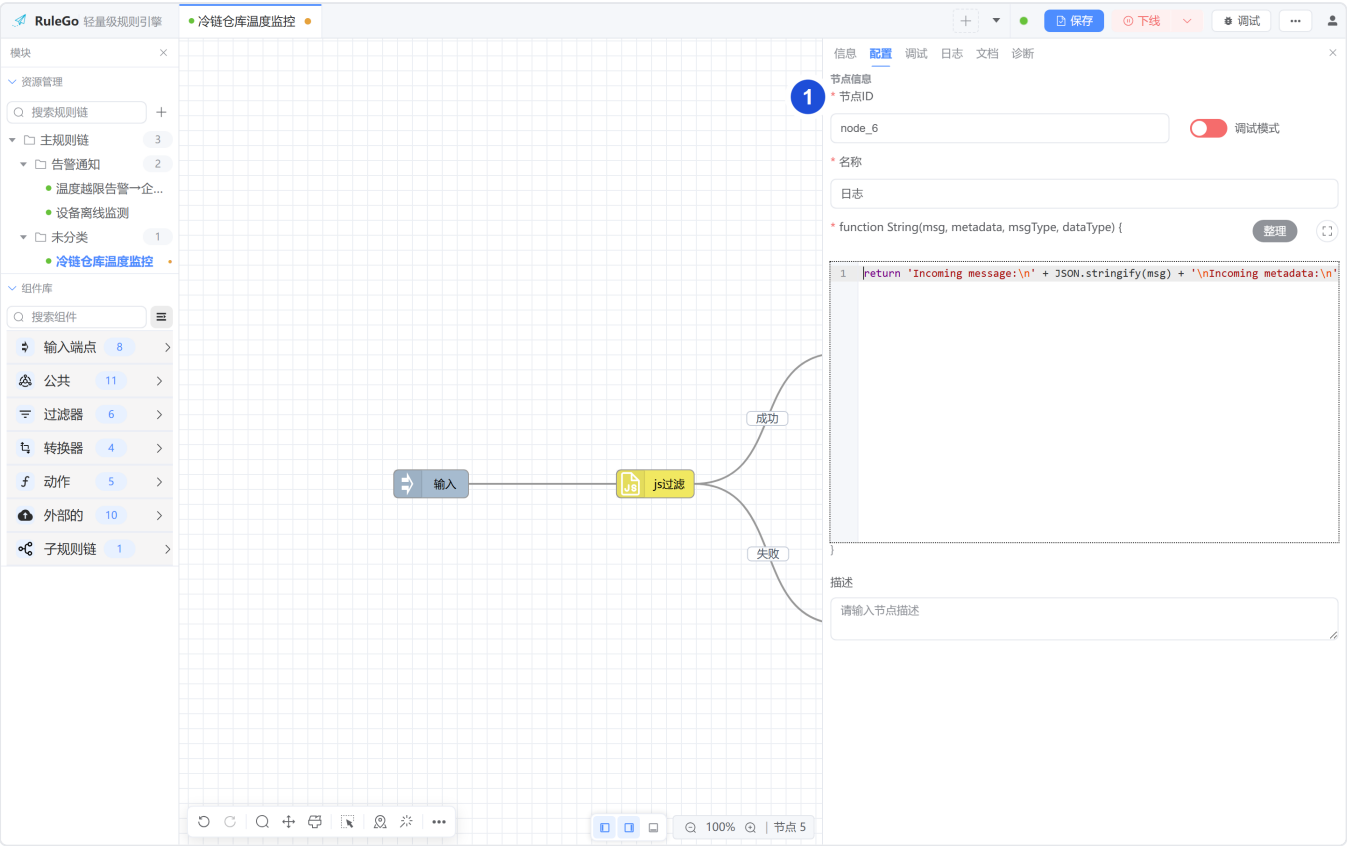
- ① **脚本编辑器**：支持 JSON 语法高亮与行号；函数签名为 `Filter(msg, metadata, msgType, dataType)`，直接写函数体即可，如 `return msg.temperature > 8`。

同样双击「js转换」节点，在其脚本中组装告警字段：

```
msg.alarm = '冷库超温：' + msg.temperature + '℃'
return msg;
```

5.3 普通表单节点

「日志」节点这类简单组件以表单呈现：

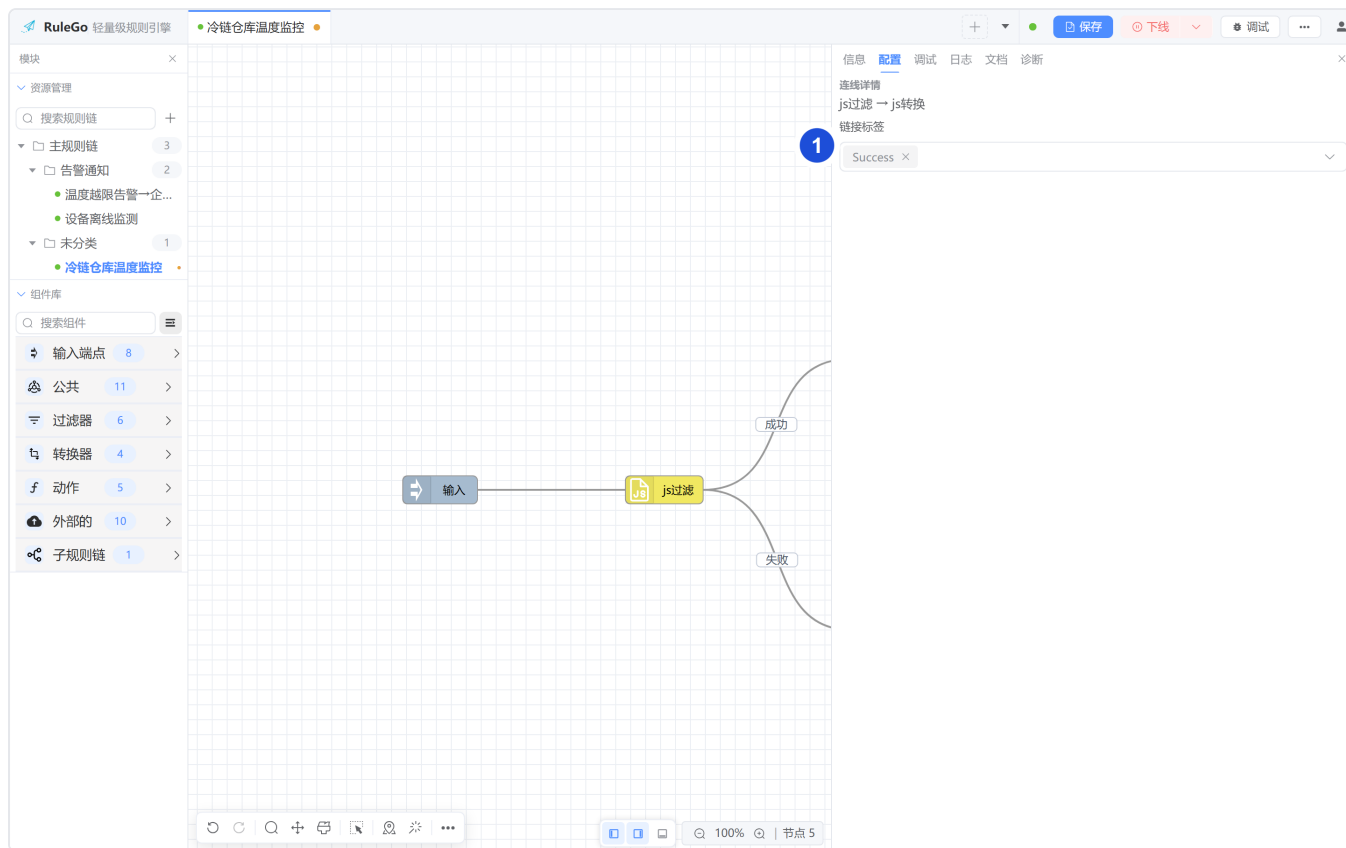


- ① **表单字段**：如是否打印到调试日志、日志内容模板等；必填项与格式校验即时提示。

部分组件的低频字段（超时、证书、高级选项）收在「高级配置」折叠组里，按需展开。

5.4 连线配置

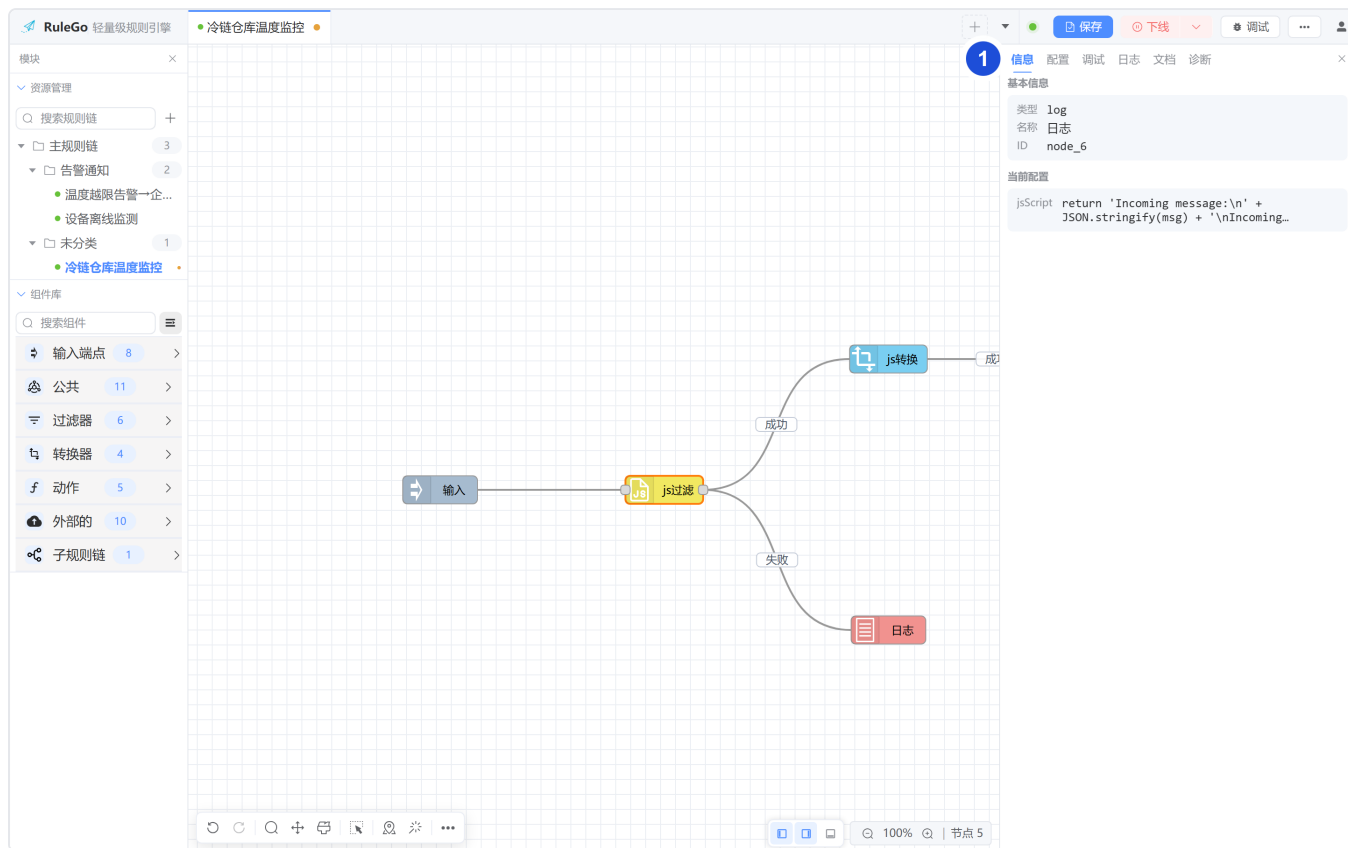
单击连线同样在右侧面板查看与修改其承载的关系标签：



- ① **链接标签**：显示当前连线的关系类型，可在这里改绑（如从「失败」改为「成功」）。

5.5 信息速览

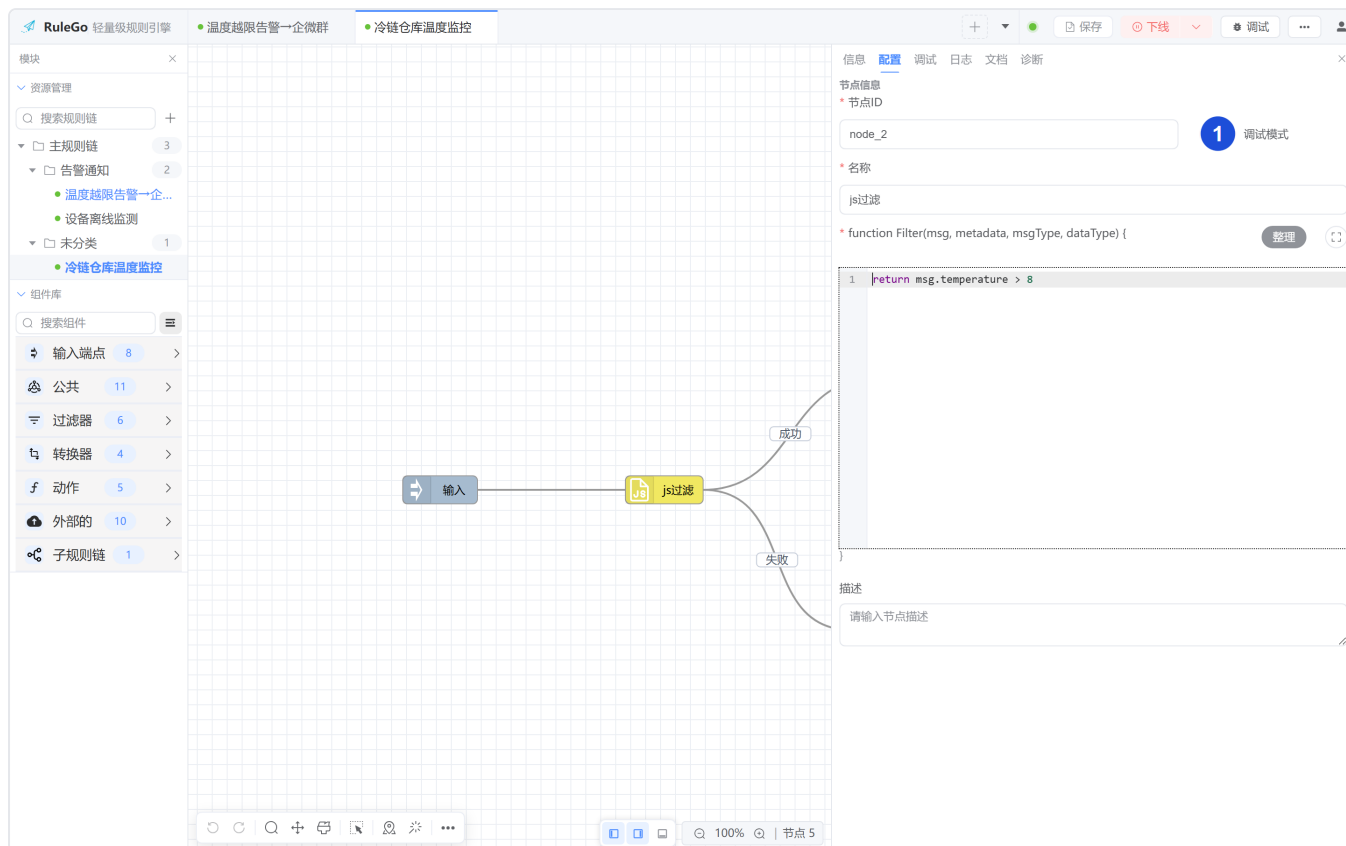
选中任意元素后，「信息」标签给出只读速览：



- ① **「信息」标签**：节点的 ID、类型、连线关系与当前配置一览，适合快速核对而无意改动。

5.6 调试模式

配置面板上方的「调试模式」开关决定该节点是否记录逐节点日志：



- ① **调试模式**：开启后，该节点每次消息进出都会留下 IN/OUT 日志，可在右侧「日志」标签与底部调试控制台查看（第 07 章）；关闭则不留痕。

提示：从编辑器调试面板发送的消息默认就是调试模式，无需预先打开开关；由外部触发（REST API、端点组件）的运行要留节点日志，要么打开对应节点的调试模式，要么在调用时带 `debugMode=true`（见第 09 章）。

5.7 节点日志、文档与诊断

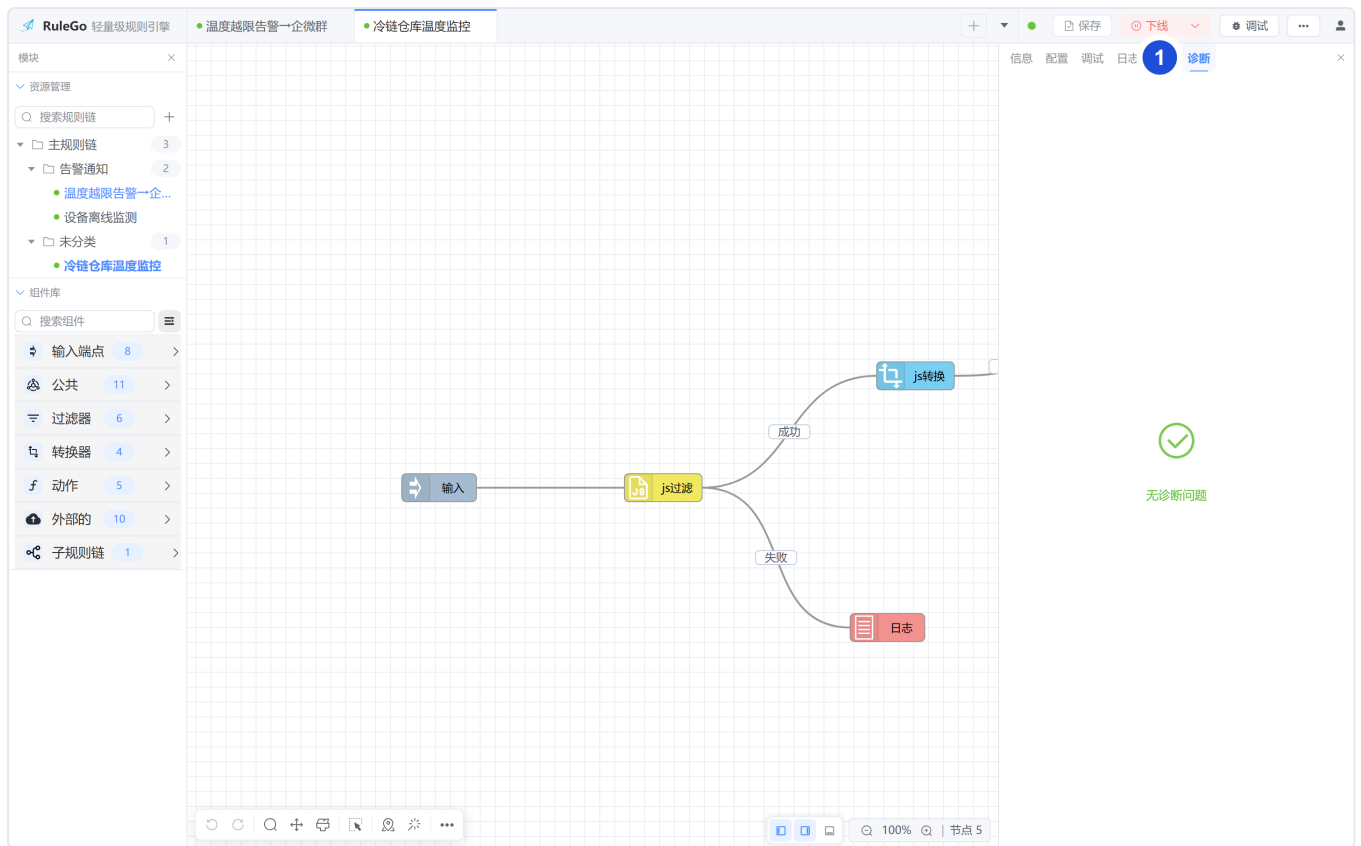
选中节点后，右侧面板还有三个实用标签：

- **日志**：该节点最近的运行日志——调试模式下逐条列出消息进出与耗时：

The screenshot displays the RuleGo Editor interface. On the left, there is a sidebar with a '模块' (Module) section containing '资源管理' (Resource Management) and '主规则链' (Main Rule Chain). Below this is a '组件库' (Component Library) with various components like '输入端点' (Input Endpoint), '公共' (Public), '过滤器' (Filter), '转换器' (Converter), '动作' (Action), '外部的' (External), and '子规则链' (Sub Rule Chain). The main workspace shows a rule chain diagram with nodes: '输入' (Input), 'js过滤' (JS Filter), '成功' (Success), '失败' (Failure), 'js转换' (JS Conversion), and '日志' (Log). The '日志' node is highlighted. On the right, a panel shows the '日志' (Log) tab with a table of log entries.

时间	类型	消息类型	关系
2026-09-20 11:06:28	IN	TEST_MSG_...	
2026-09-20 11:06:28	OUT	TEST_MSG_...	True
2026-09-20 11:01:08	OUT	TEST_MSG_...	True
2026-09-20 11:01:08	IN	TEST_MSG_...	
2026-09-20 11:01:02	IN	TEST_MSG_...	
2026-09-20 11:01:02	OUT	TEST_MSG_...	True

- ① **节点日志**：只看当前选中节点的日志，比整条链的控制台更聚焦。
- **文档**：当前组件的官方说明（参数含义、示例），不用离开编辑器查资料。
- **诊断**：整链的静态检查结果（未连线、缺配置等拓扑问题），点击条目直接定位到问题节点：

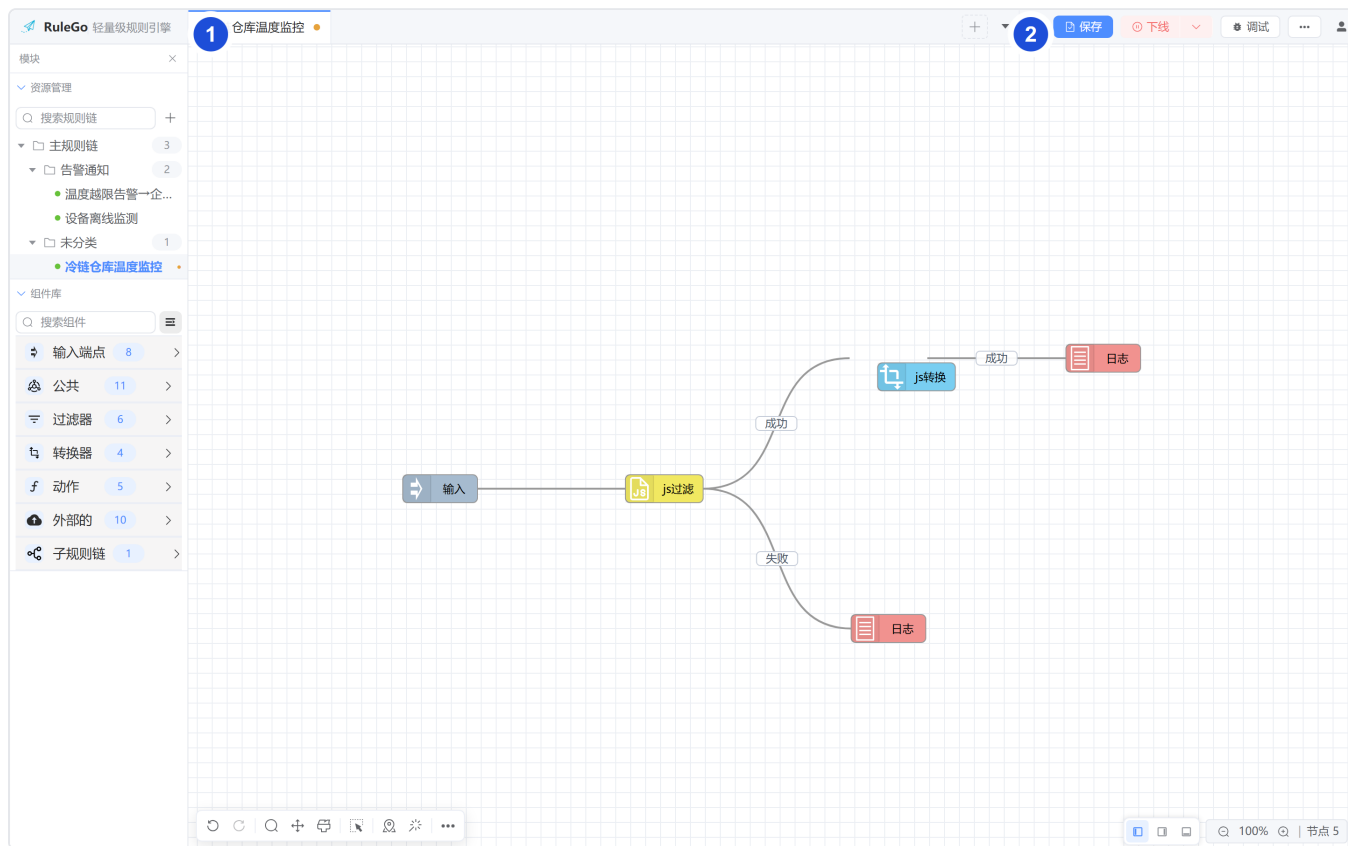


- ①「**诊断**」标签：列出整链的拓扑问题与所在节点，点击条目直接跳转；当前链没有问题，显示绿色对勾。

06 保存与部署

6.1 脏态提示

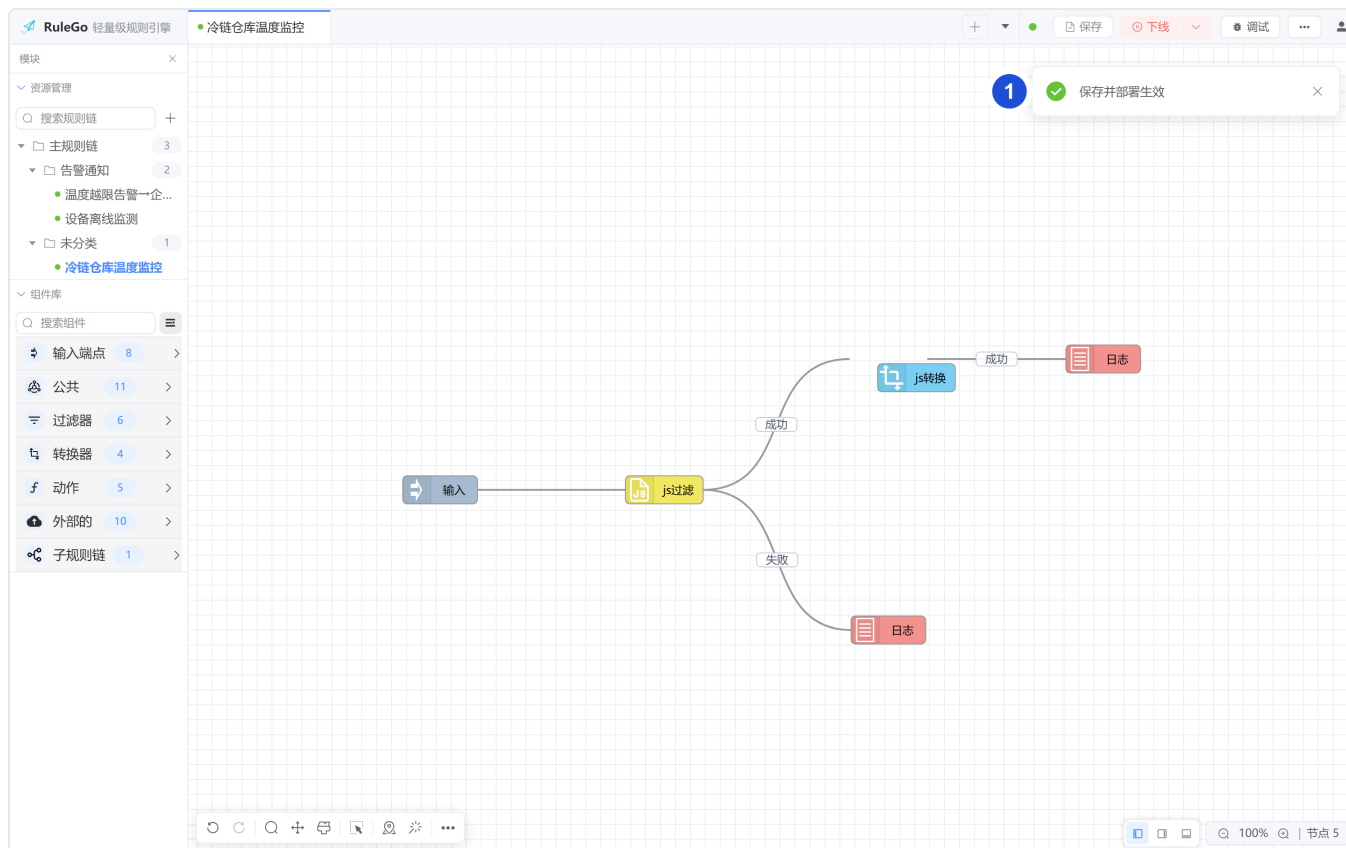
画布有任何未保存修改时，标签页标题出现**红点**，顶栏「保存」按钮变为实心可点：



- ① **标签页红点**：该链有未保存的修改。
- ② **「保存」按钮**：仅在修改时可点。

6.2 保存

点击「保存」（或快捷键 **Ctrl+S**），结果以右上角通知反馈：

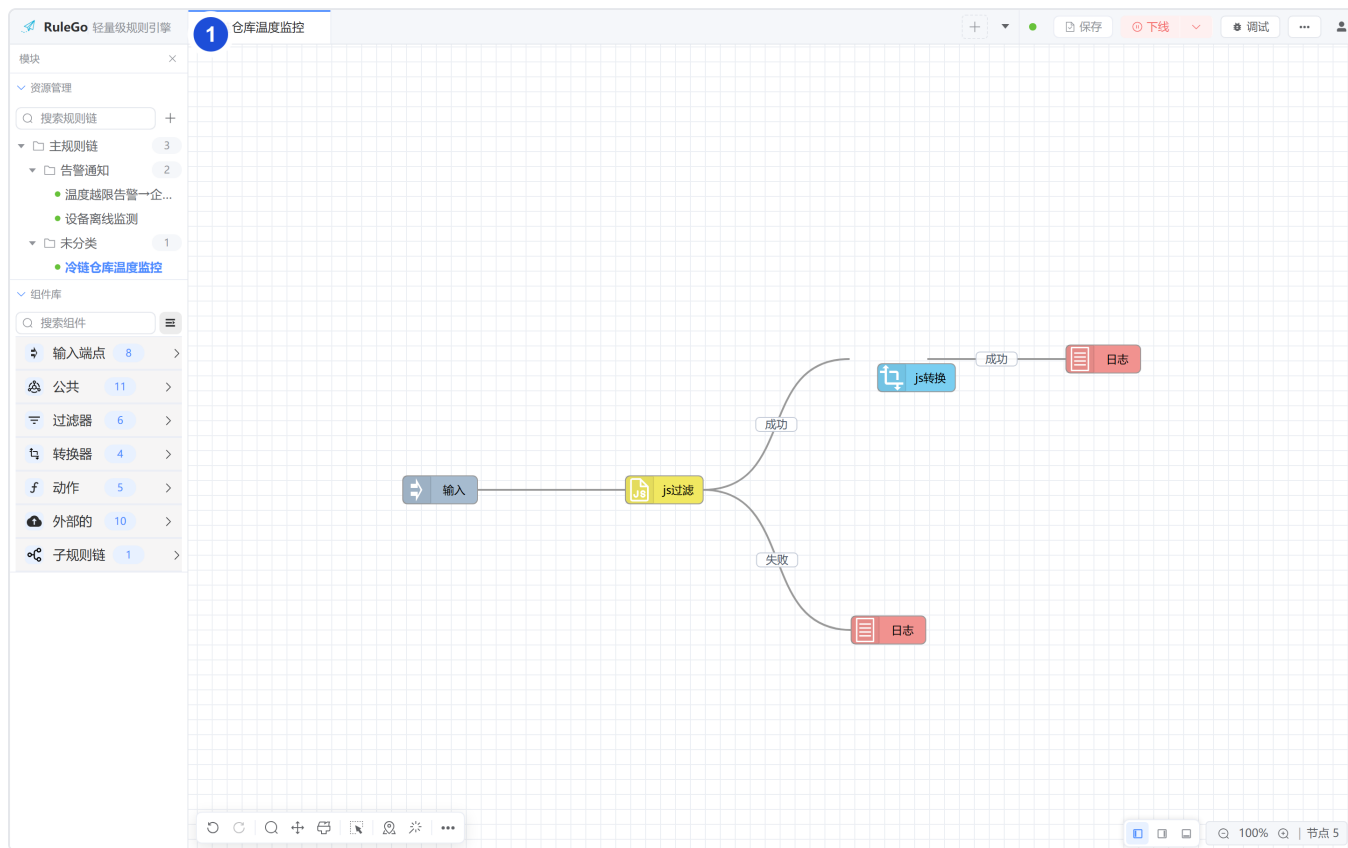


- ① **保存成功通知**：运行中的链保存后**新版本立即生效**；已下线的链保存只落库、保持下线，通知里会附带「部署」按钮引导上线。

保存前若服务端版本已被他人修改，会弹出冲突确认，避免覆盖别人的改动。

6.3 运行状态徽标

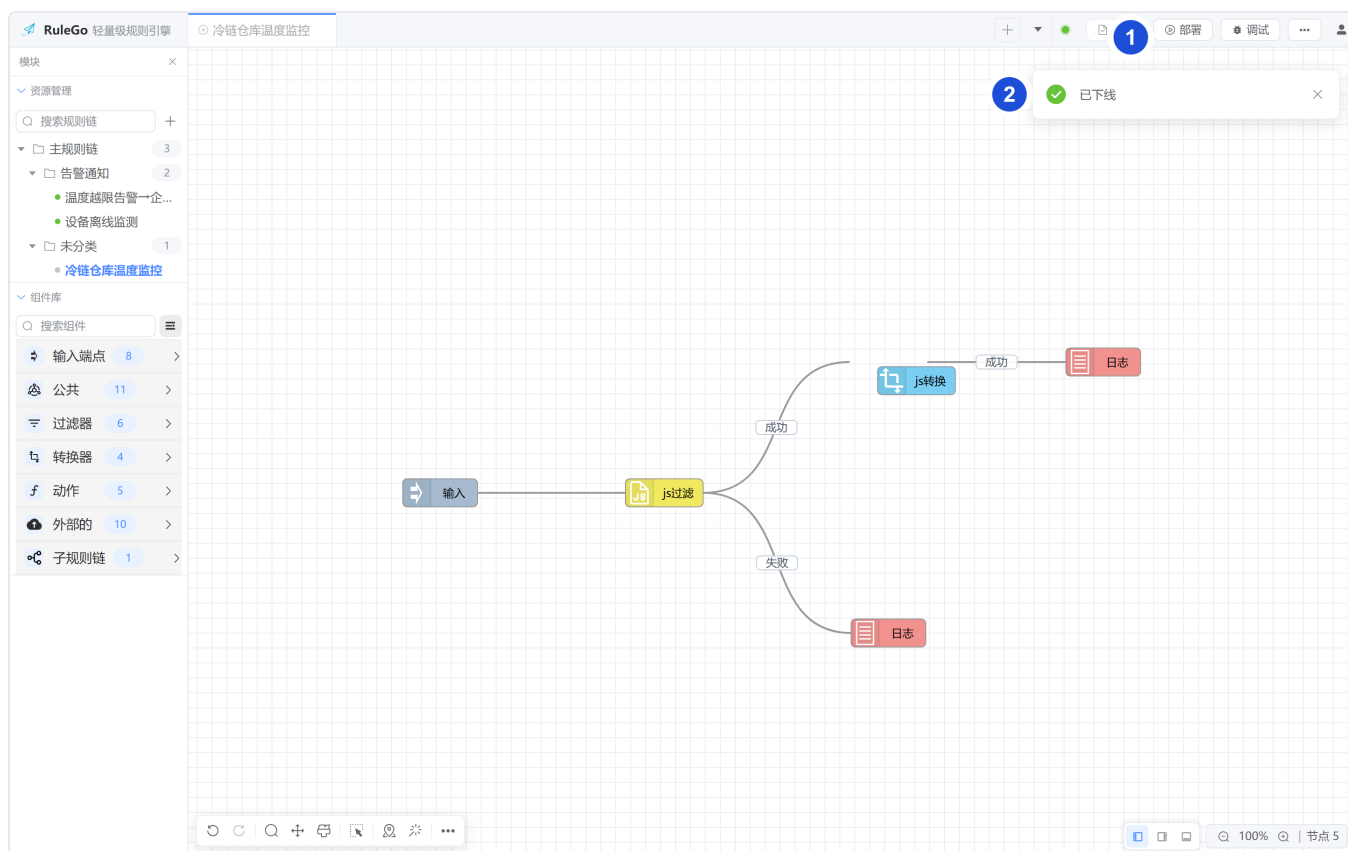
标签页与链列表中的徽标反映每条链的运行状态：



- ① **运行中徽标**：绿色表示链在引擎中运行；灰色/划线表示已下线；服务端不存在的草稿链无徽标。

6.4 下线与重新部署

顶栏的运行开关按当前状态变化：运行中显示红色「下线」，已停则显示「部署」。下线一条干净的链即时生效：



- ① 「部署」按钮：下线后顶栏切换为部署入口，点击重新上线。
- ② 已下线通知：右上角确认操作结果。

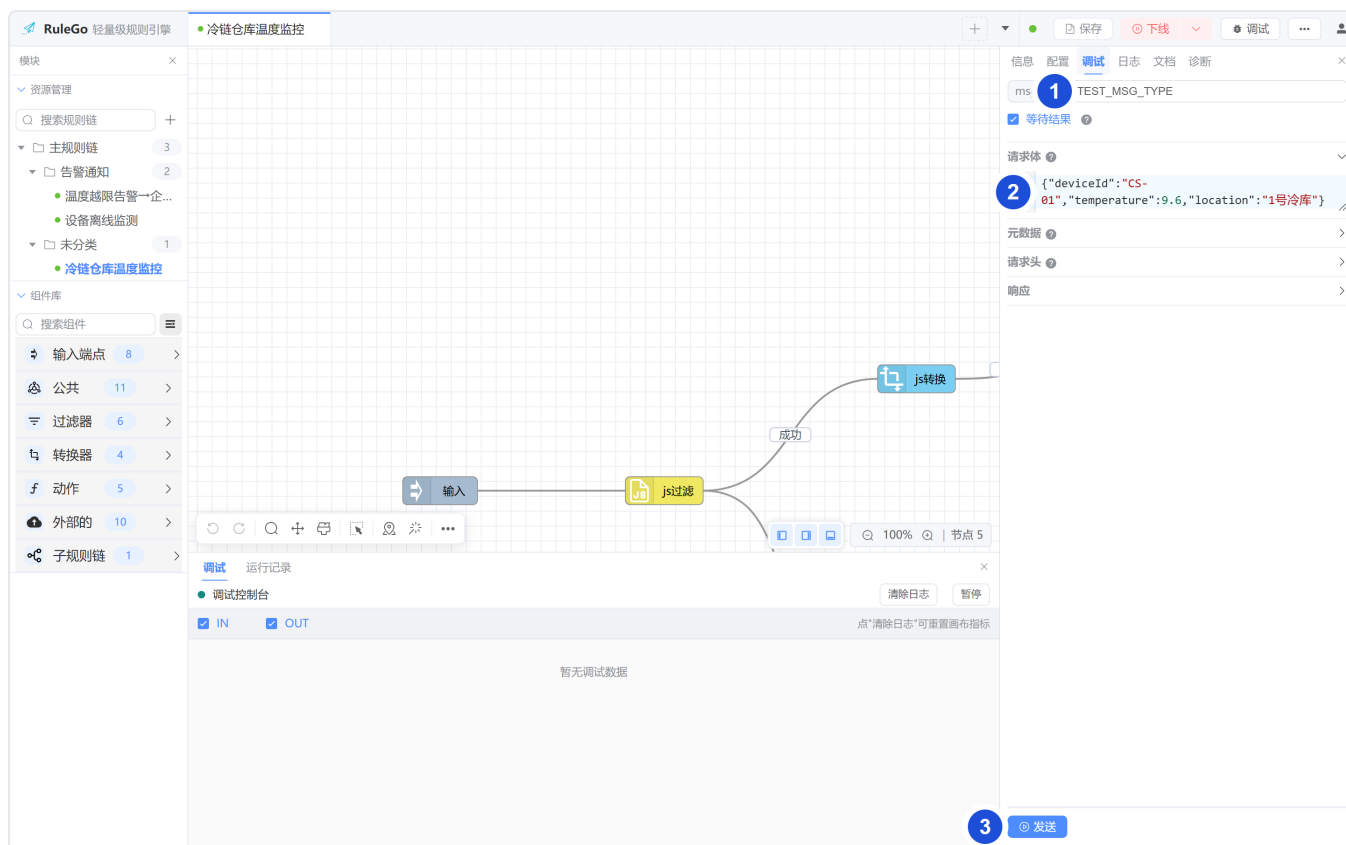
「下线」按钮旁的下拉还提供**重启此链 / 重启全部**（重启全部指当前已打开标签页的链），用于节点内存态复位。

术语约定：**部署** = 让改动上线运行；**下线** = 停止规则链。保存永远只是保存，不会改变运行状态。

07 调试运行

7.1 调试面板

点击顶栏「调试」按钮，右侧面板切到「调试」标签。填入一条模拟的冷库高温消息：



- ① **消息类型**：模拟的消息类型，默认 `TEST_MSG_TYPE`，可用 `msgTypeSwitch` 类节点按类型分流。
- ② **请求体**：消息主数据，JSON 格式；节点里用 `msg.字段名` 或 `${msg.字段名}` 取值。
- ③ **发送**：把消息发往当前规则链执行；勾选「等待结果」同步取回出口消息。

7.2 发送与查看结果

点击「发送」，面板下方立即显示执行响应：

The screenshot displays the RuleGo Editor interface for a rule chain named "冷链仓库温度监控". The rule chain is composed of three nodes: "输入" (Input), "js过滤" (JS Filter), and "js转换" (JS Transform). The "js过滤" node is highlighted, and its configuration is shown in the right panel. The "js转换" node is also highlighted, and its configuration is shown in the right panel. The bottom panel shows the "调试" (Debug) tab with a "运行记录" (Run Log) table.

js过滤 (node_2) Configuration:

```
1 {  
2   "deviceId": "CS-01",  
3   "temperature": 9.6,  
4   "location": "1号冷库"  
5 }
```

js转换 Configuration:

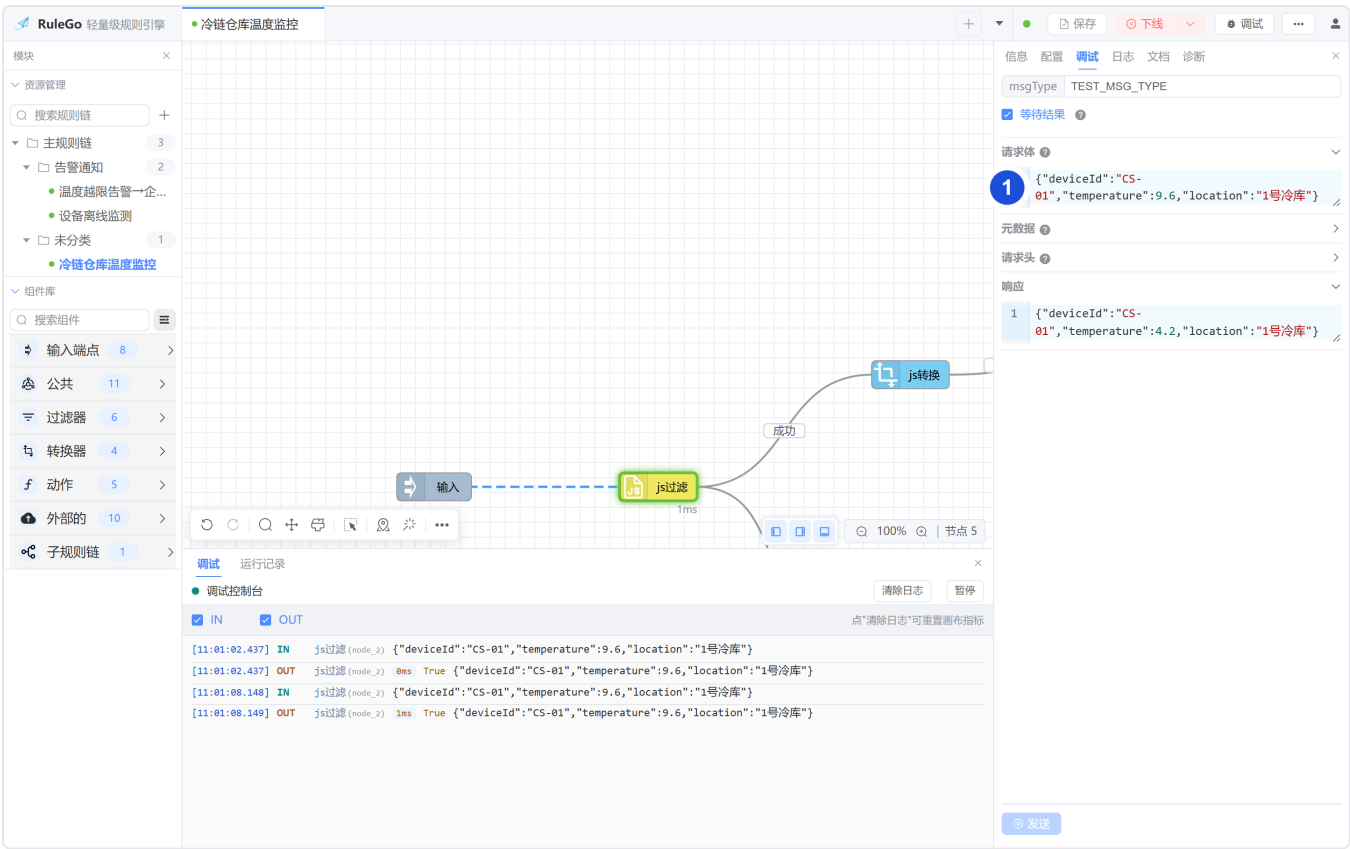
```
1 {  
2   "deviceId": "CS-01",  
3   "temperature": 9.6,  
4   "location": "1号冷库"  
5 }
```

运行记录 (Run Log) Table:

IN	OUT
[11:01:02.437] IN	js过滤 (node_2) {"deviceId": "CS-01", "temperature": 9.6, "location": "1号冷库"}
[11:01:02.437] OUT	js过滤 (node_2) 0ms True {"deviceId": "CS-01", "temperature": 9.6, "location": "1号冷库"}

- ① **消息体编辑器**：可反复修改重发，模拟不同工况。
- ② **响应区**：显示链出口消息、元数据与响应头——高温消息被 js 过滤判定为真，沿「成功」分支经 js 转换组装出告警字段。

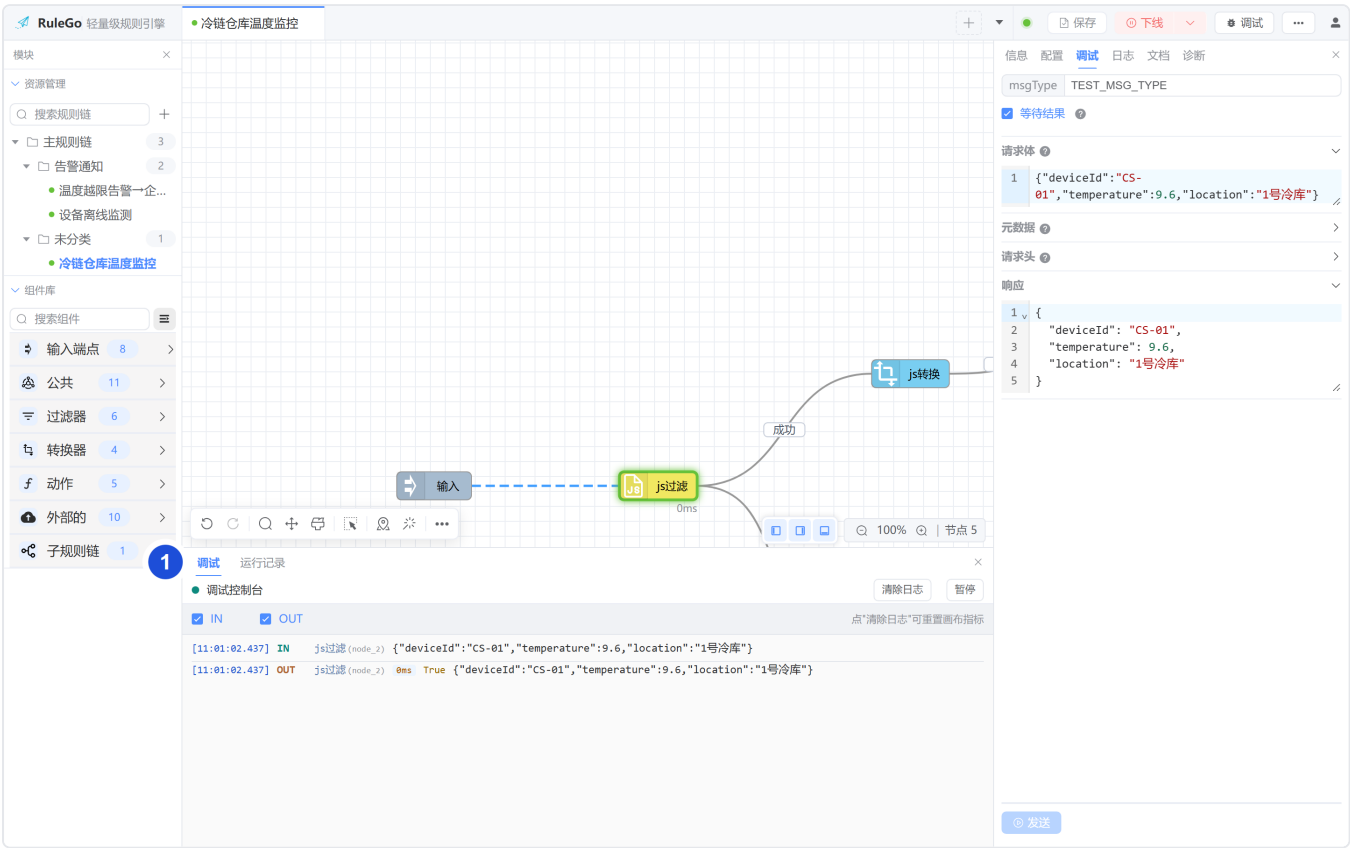
换成正常温度再发一条：



- ① 消息体编辑器： temperature: 4.2 未超阈值，消息走「失败」分支记录正常温度日志。

7.3 调试控制台

要看**逐节点**的消息流动，先打开底部面板的「调试」控制台（画布左下工具栏的底部面板开关，或画布右键 → 面板 → 调试），再发送消息——控制台通过 WebSocket 实时接收每个节点的进出消息：

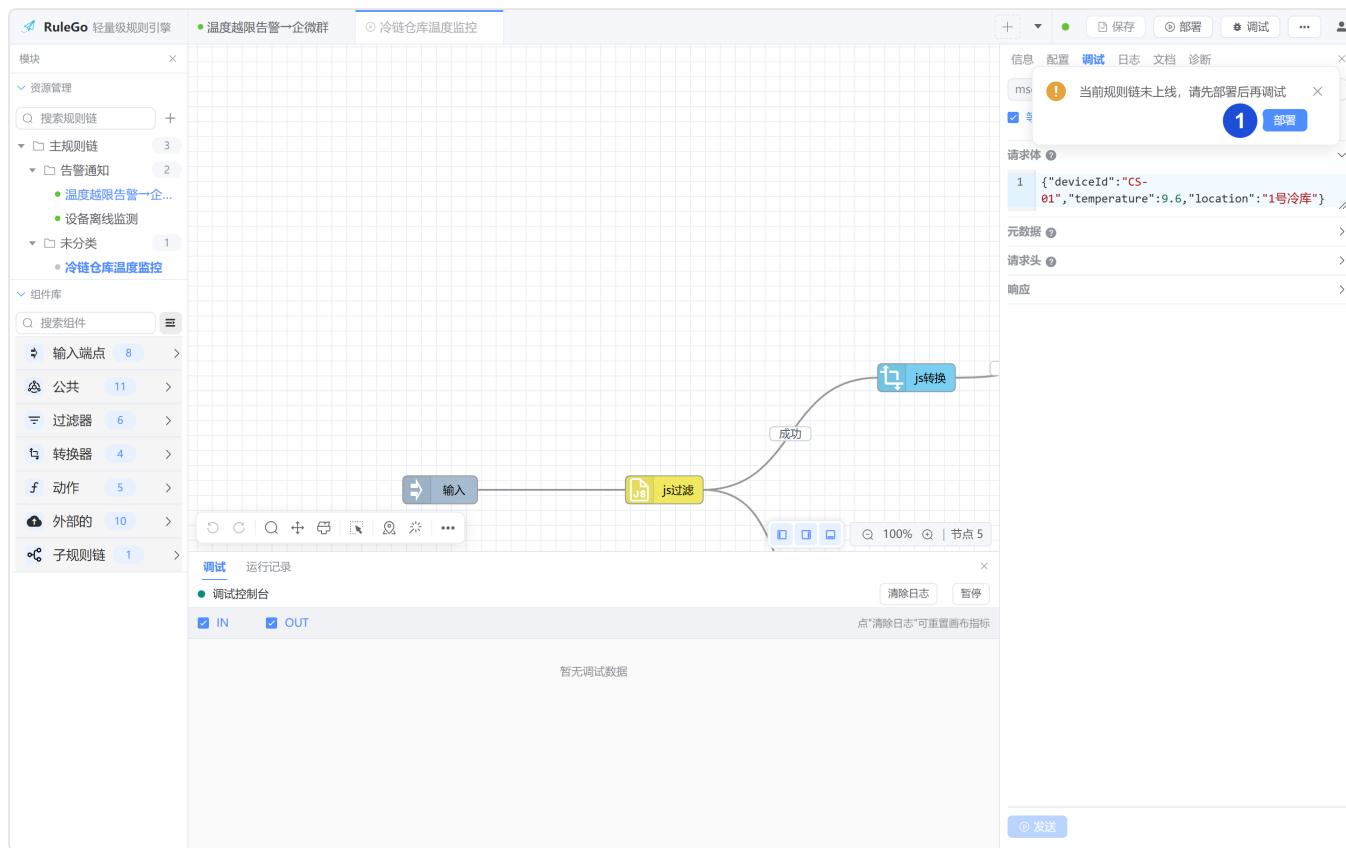


- ① **调试控制台**：按时间序列出每个节点的 IN（进入消息）与 OUT（流出消息及耗时），可逐节点核对数据被哪一步改写；支持暂停滚动与清除日志。

逐节点日志只在**调试模式**下记录：编辑器发送的调试消息默认带调试模式，开箱即有日志；外部触发想留痕则见第 05、09 章。只想看某一个节点的日志时，右侧面板的「日志」标签更聚焦（第 05 章）。

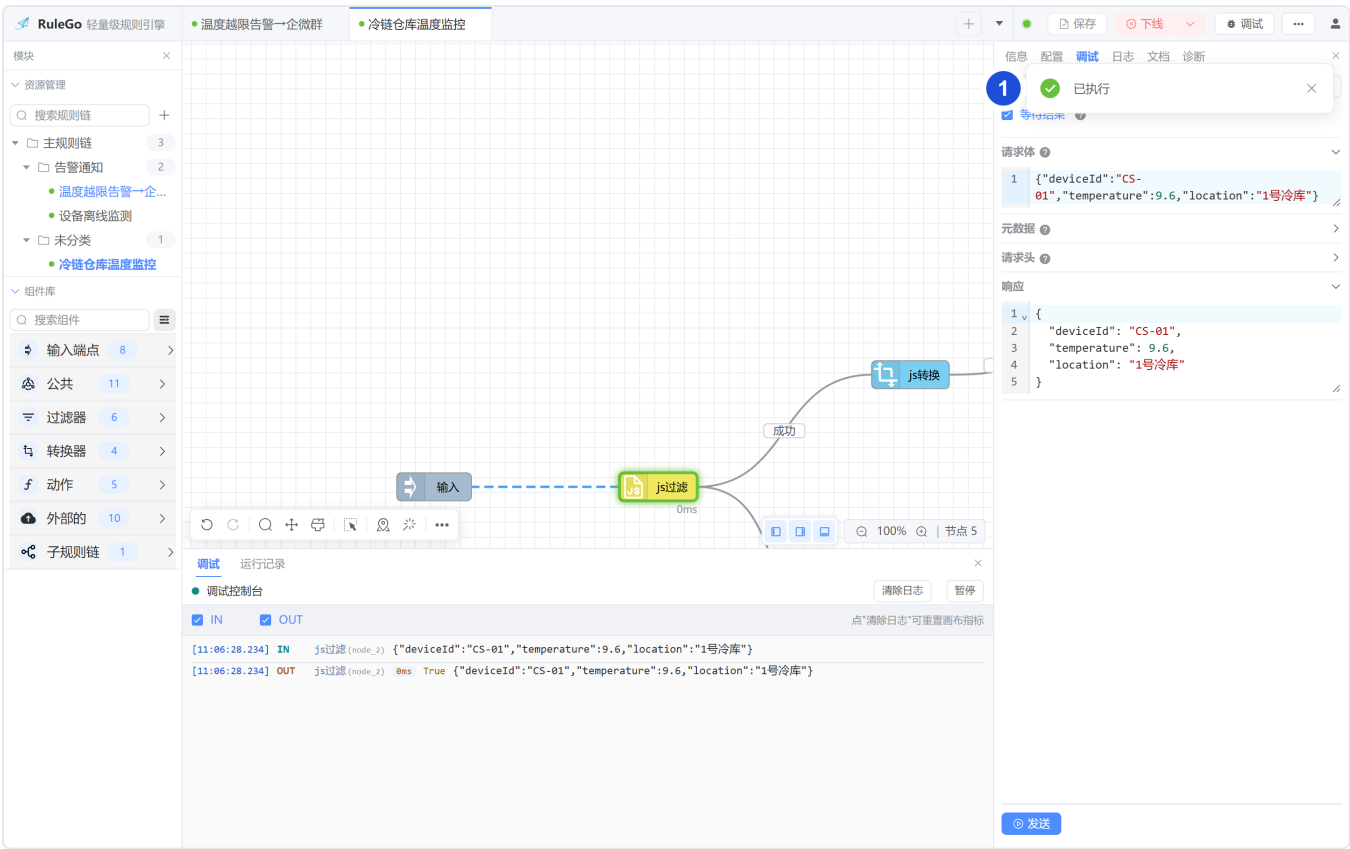
7.4 未运行链的调试守卫

对已下线（或尚未部署）的链发送消息会被拦截，并给出带「部署」按钮的引导通知：



- ① **部署引导通知：** 点击「部署」直接上线该链。

部署完成后，被拦截的那条消息会**自动重发一次**，无需手动再点：



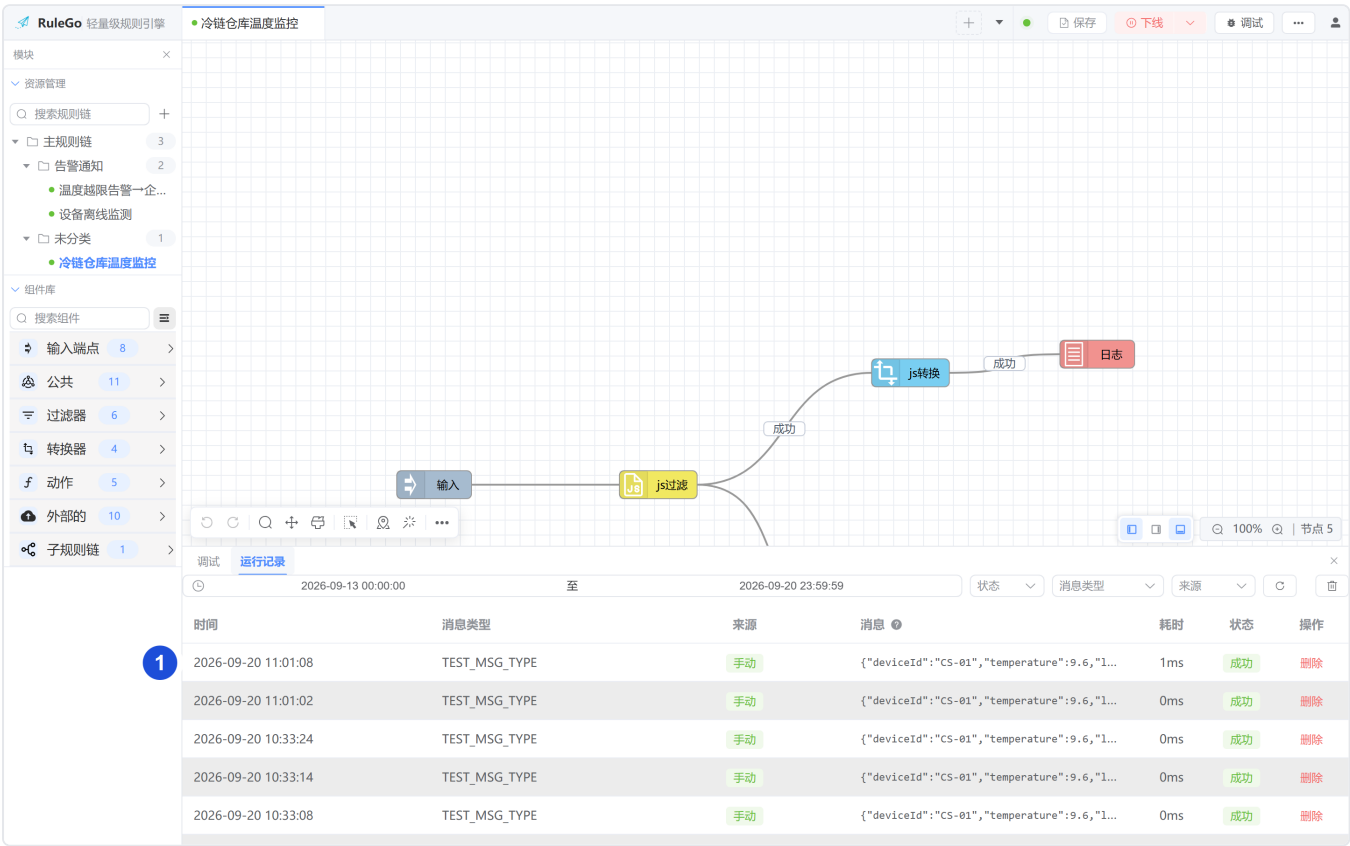
- ① **执行通知**：部署成功并自动完成重发，调试流程不中断。

提示：「从节点起跑」「仅该节点」选项可把调试范围缩小到单个节点，定位问题更精准；画布上节点的右键菜单也带「从此节点运行」。

08 运行记录

8.1 记录列表

每次规则链执行（无论手动调试还是外部触发）都会自动留痕。打开底部面板的「运行记录」标签：



- ① **执行记录行**：每行显示触发时间、消息类型、来源（手动 / HTTP / 定时等）、消息摘要、耗时与成败状态；右侧「删除」可清理单条记录。

顶部支持按时间段与状态筛选。第 07 章的两次调试发送在这里一目了然。

8.2 执行详情

点击任意一行查看该次执行的详情：

The screenshot displays the RuleGo Editor interface. The top bar shows the title 'RuleGo 轻量级规则引擎' and the current rule chain '冷链仓库温度监控'. The left sidebar contains a '模块' (Modules) section with '资源管理' (Resource Management) and '主规则链' (Main Rule Chain) categories. The '主规则链' section lists several rule chains, including '冷链仓库温度监控'. The main workspace shows a workflow diagram with nodes: '输入' (Input), 'js过滤' (JS Filter), 'js转换' (JS Transform), and '日志' (Log). The workflow is connected by lines, and the status '成功' (Success) is shown next to the 'js转换' node. The bottom panel shows the '运行记录' (Execution Log) for the selected rule chain. It includes a '返回列表' (Return List) button and a table with columns: '节点' (Node), '关系类型' (Relationship Type), '耗时' (Duration), '状态' (Status), and '操作' (Action). The table currently shows '暂无数据' (No data).

① 返回列表：回到记录列表。

② 节点明细表：显示该次执行经过的节点、关系类型、耗时与状态；点击节点的「消息」可展开该节点收发的完整数据。

提示：节点级明细需要在链信息中把「运行记录级别」设为**详情**（默认摘要级只记录链出口消息）。链级设置路径：标签页右键 → 编辑信息 → 运行记录级别。

09 触发器与集成

规则链搭好后，需要有「入口」让消息流进来。本章介绍三种触发方式的选择、端点组件的用法，以及平台内置的 REST/AI/MCP 集成接口。

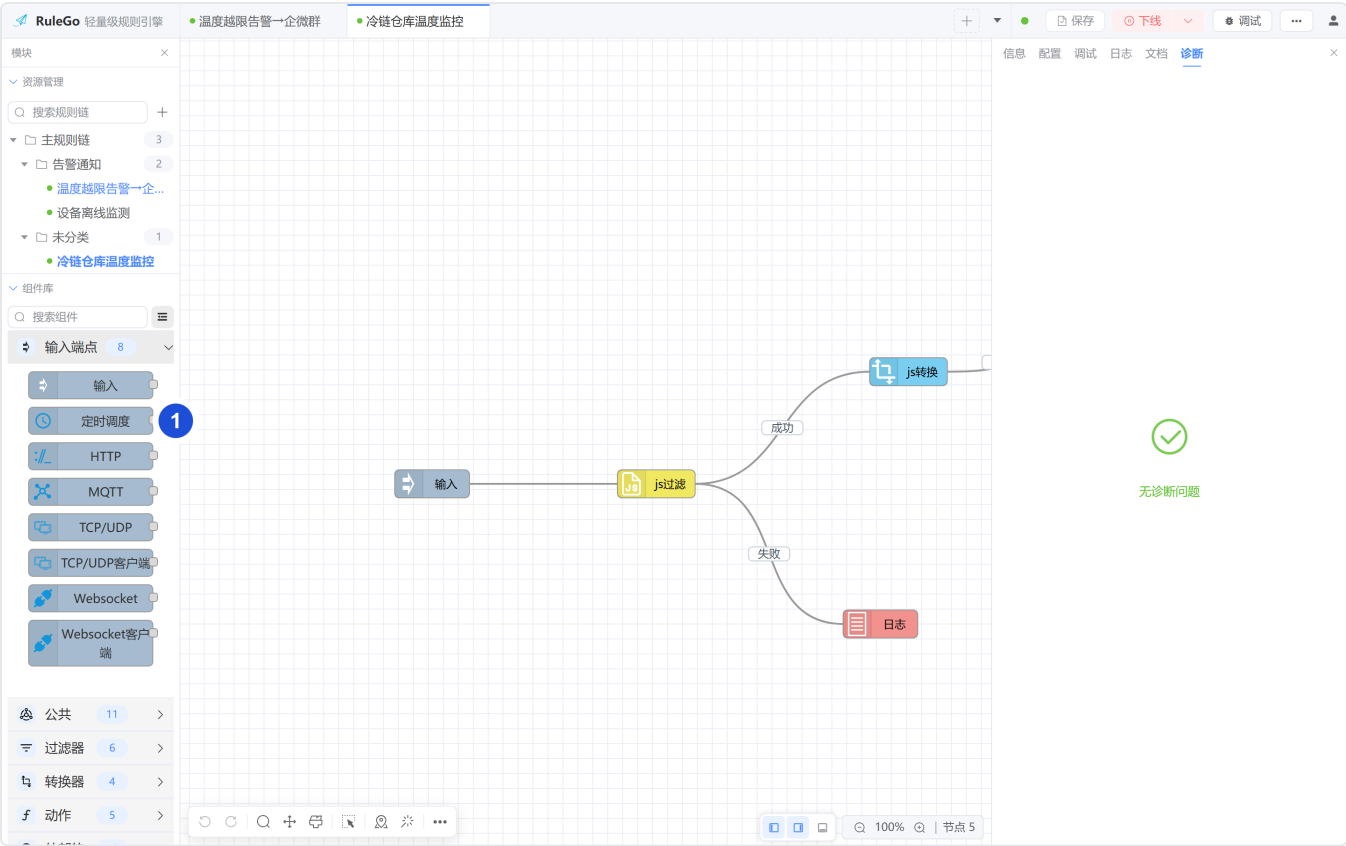
9.1 触发方式怎么选

场景	推荐方式	说明
外部系统发 HTTP 请求触发	内置 REST API (见 9.4)	直接 <code>POST /api/v1/rules/{链ID}/execute/...</code> ，零配置
定时周期执行	定时调度端点 (见 9.3)	Cron 表达式驱动
MQTT/Kafka 等消息中间件接入	对应输入端点组件 (见 9.2)	订阅主题/消费 topic 后流入链
把链当 AI 工具/对话服务用	OpenAI 兼容接口 / MCP (见 9.5、9.6)	供 AI 客户端与智能体平台调用

提示：**HTTP 触发优先用内置 REST API**，不要为了「让外部调一下」专门新建一个 HTTP 端点节点——内置接口鉴权、运行日志、消息类型路由都已就绪。HTTP 端点节点适合做协议边缘接入（自定义路径路由、Webhook 适配、报文转换等）。

9.2 输入端点组件

组件库的「输入端点」分组是所有触发型组件的集合。点击分组名展开：



- ① **定时调度**：按 Cron 周期触发规则链，是最常用的触发器。

常用端点组件一览：

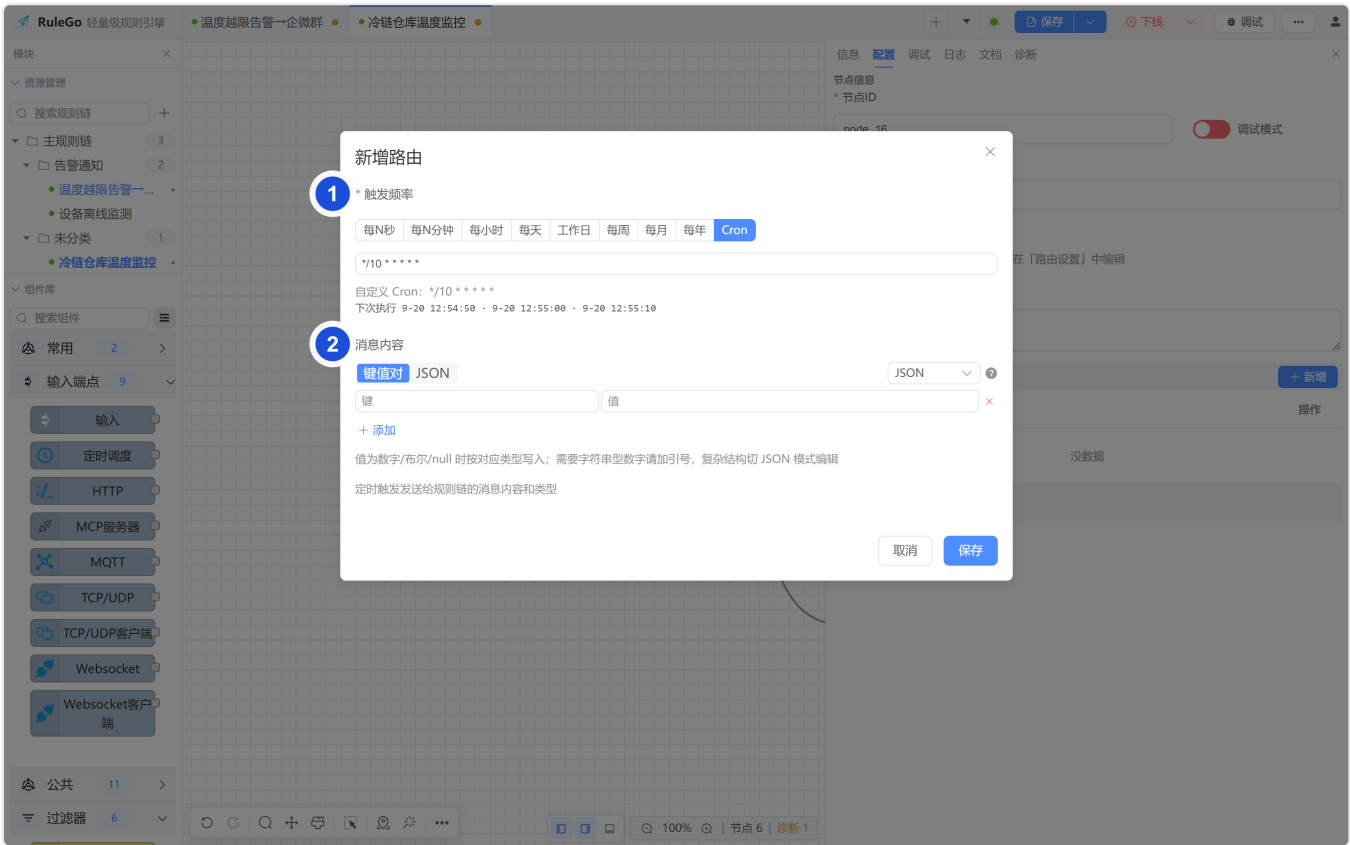
组件	用途
定时调度	Cron 表达式周期触发
MQTT	订阅 MQTT 主题，收到消息即触发
HTTP	监听自定义路径的 HTTP 请求
WebSocket / WS 客户端	WS 服务端监听 / 主动连接外部 WS
TCP/UDP	监听 TCP/UDP 端口收原始报文
Kafka / NATS / RabbitMQ / Redis / Pulsar / NSQ	消费对应中间件的消息
MySQL CDC	订阅数据库变更事件
MCP Server	把当前链暴露为 MCP 工具

把端点组件拖入画布，它就是链的触发入口；再从它的输出锚点连线到处理节点。选中端点节点后，配置面板会多出「路由设置」标签，用于配置消息的路由与转发规则。

端点配置完成并保存后即时生效（服务端「保存即部署」）。要停止触发，把链**下线**即可（见第 06 章）。

9.3 定时触发

拖入「定时调度」组件，在右侧配置面板的「路由设置」中点「新增」，配置一条触发路由：

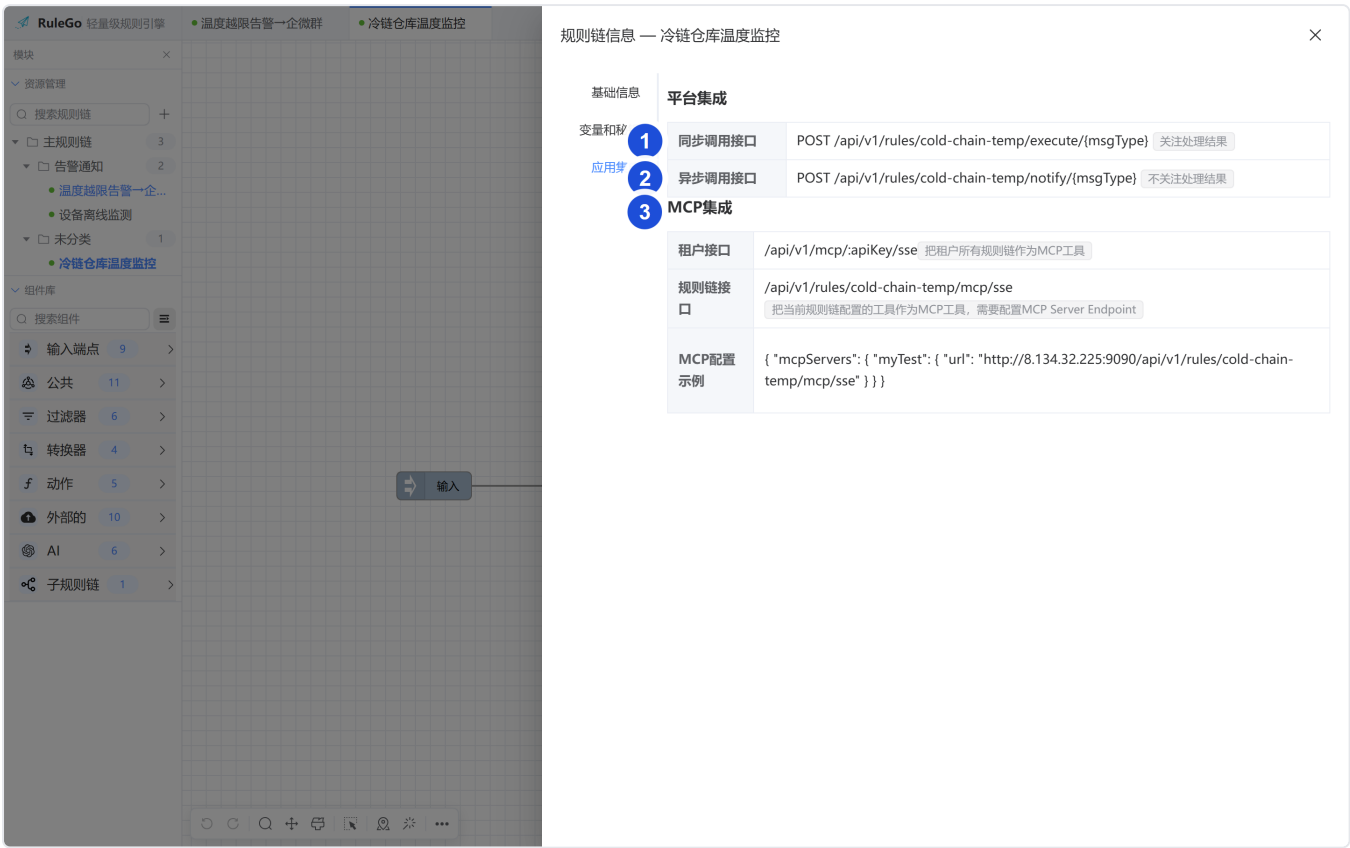


- ① **触发频率**：点选常用周期（每 N 分钟、每天、工作日等），或切到 Cron 自定义表达式（如 `0 */5 * * * *` 每 5 分钟）；下方实时预览下次执行时间。
- ② **消息内容**：每次定时触发发给规则链的消息体，支持键值对或 JSON，后续节点从 `msg` 中取值。

定时消息流经后续节点做处理——典型的「定时采集」用法是：定时调度 → 读点位/查数据 → 条件判断 → 告警或记录。配置保存并上线后即按周期执行。

9.4 内置 REST API 触发

每条规则链天生自带两个执行接口，无需任何配置。入口：顶栏「…」更多操作菜单 → **规则链** → **编辑信息** → **应用集成**标签：



- ① **同步调用接口**：等待规则链执行完毕并返回结果。
- ② **异步调用接口**：触发后立即返回，不等待执行结果。
- ③ **MCP 集成**：把链暴露为 MCP 工具（见 9.6）；包含 ai/agent 节点的链另有 OpenAI 兼容对话接口（见 9.5）。

接口格式（`{msgType}` 是消息类型，自定义字符串，供 `msgTypeSwitch` 等节点按类型分流）：

```
# 同步执行：等待链执行完，响应里带出口消息
curl -X POST http://localhost:9090/api/v1/rules/cold-chain-temp/execute/telemetry \
  -H "Content-Type: application/json" \
  -d '{"deviceId":"CS-01","temperature":9.6,"location":"1号冷库"}'

# 异步执行：高吞吐场景，触发即返回
curl -X POST http://localhost:9090/api/v1/rules/cold-chain-temp/notify/telemetry \
  -H "Content-Type: application/json" \
  -d '{"deviceId":"CS-01","temperature":9.6}'
```

开启鉴权时（`require_auth = true`，见第 14 章）携带凭证：

```
-H "X-API-Key: ak-xxxx"           # 方式一：API Key
-H "Authorization: Bearer <token>" # 方式二：登录令牌
```

可用的查询参数：

参数	说明
<code>msgId</code>	自定义消息 ID，便于链路与运行记录对账
<code>debugMode=true</code>	按调试模式执行，记录逐节点日志（运行记录里来源标记为「手动」）
<code>_fromNodeId</code>	指定从某个节点开始执行
<code>_onlyNodeId</code>	只执行指定节点

提示：通过内置接口的每次执行都会进入[运行记录](#)（第 08 章），排查线上问题先看那里。

9.5 OpenAI 兼容接口

链里包含 **AI 智能体 (ai/agent)** 节点时，该链自动获得一个 OpenAI Chat Completions 兼容端点，可被任何支持 OpenAI 协议的客户端直接调用：

```
curl -X POST http://localhost:9090/api/v1/rules/{链ID}/v1/chat/completions \
-H "Content-Type: application/json" \
-d '{"model": "rulego-agent", "messages": [{"role": "user", "content": "你好"}]}'
```

支持流式响应（`"stream": true`）。智能体链的搭建见第 12 章。

9.6 MCP 集成

RuleGo Server 内置 MCP 服务，可以把规则链当作 MCP 工具暴露给 Claude、Cursor 等 AI 客户端：

- **租户接口** `POST /api/v1/mcp/{apiKey}`：把当前账号下**所有**规则链作为工具集暴露（StreamableHTTP）。
- **单链接口**：仅暴露某一条链；配置示例在「应用集成」标签里可直接复制：

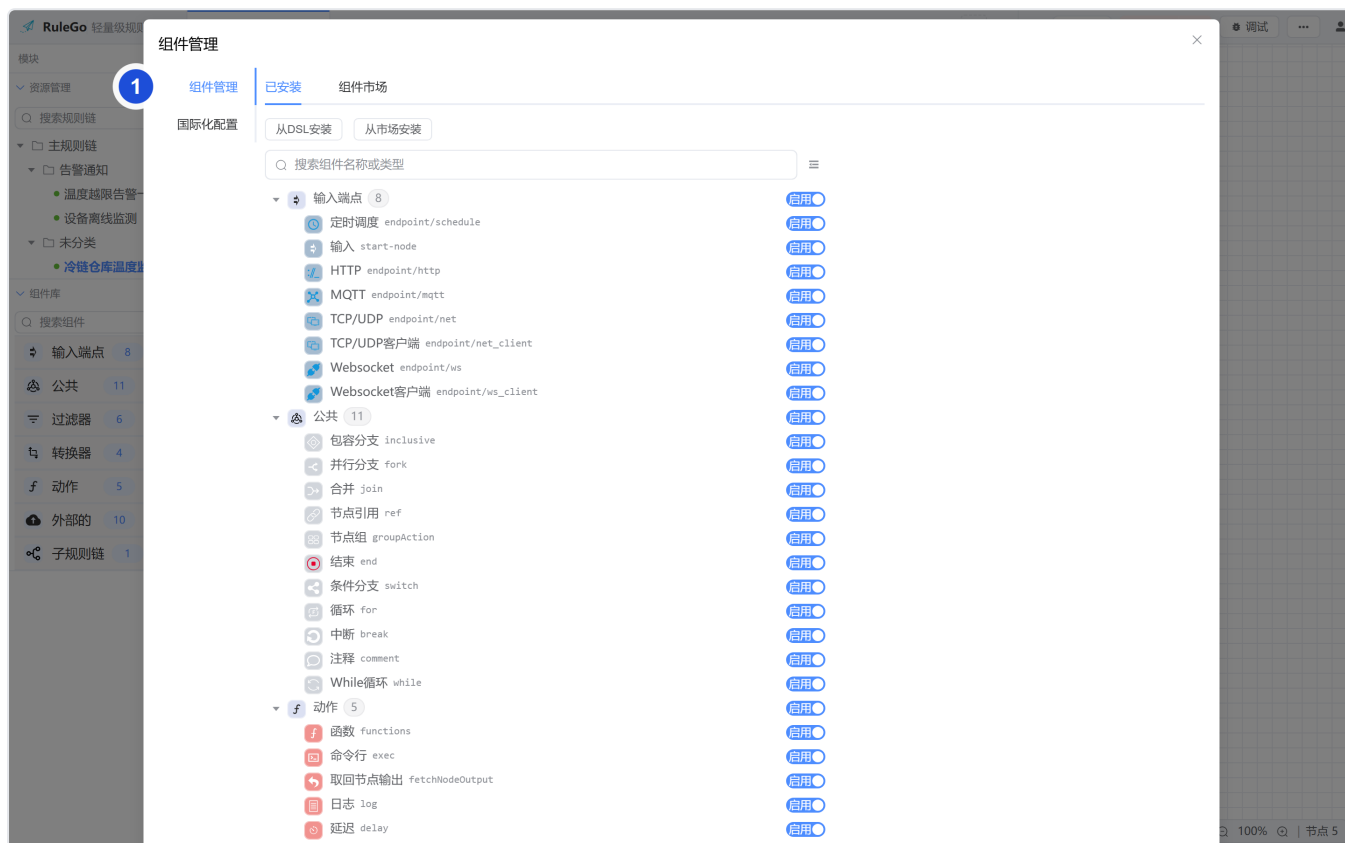
```
{
  "mcpServers": {
    "rulego": {
      "url": "http://your-rulego-server/api/v1/mcp/YOUR_API_KEY"
    }
  }
}
```

按分组过滤工具子集用 `POST /api/v1/mcp/{apiKey}/group/{groupName}` 。

10 组件、子规则链与共享节点

10.1 组件管理

除画布直接可用的内置组件外，还可以安装动态组件扩展能力。入口：顶栏「…」更多操作菜单 → 规则链 → 组件管理：

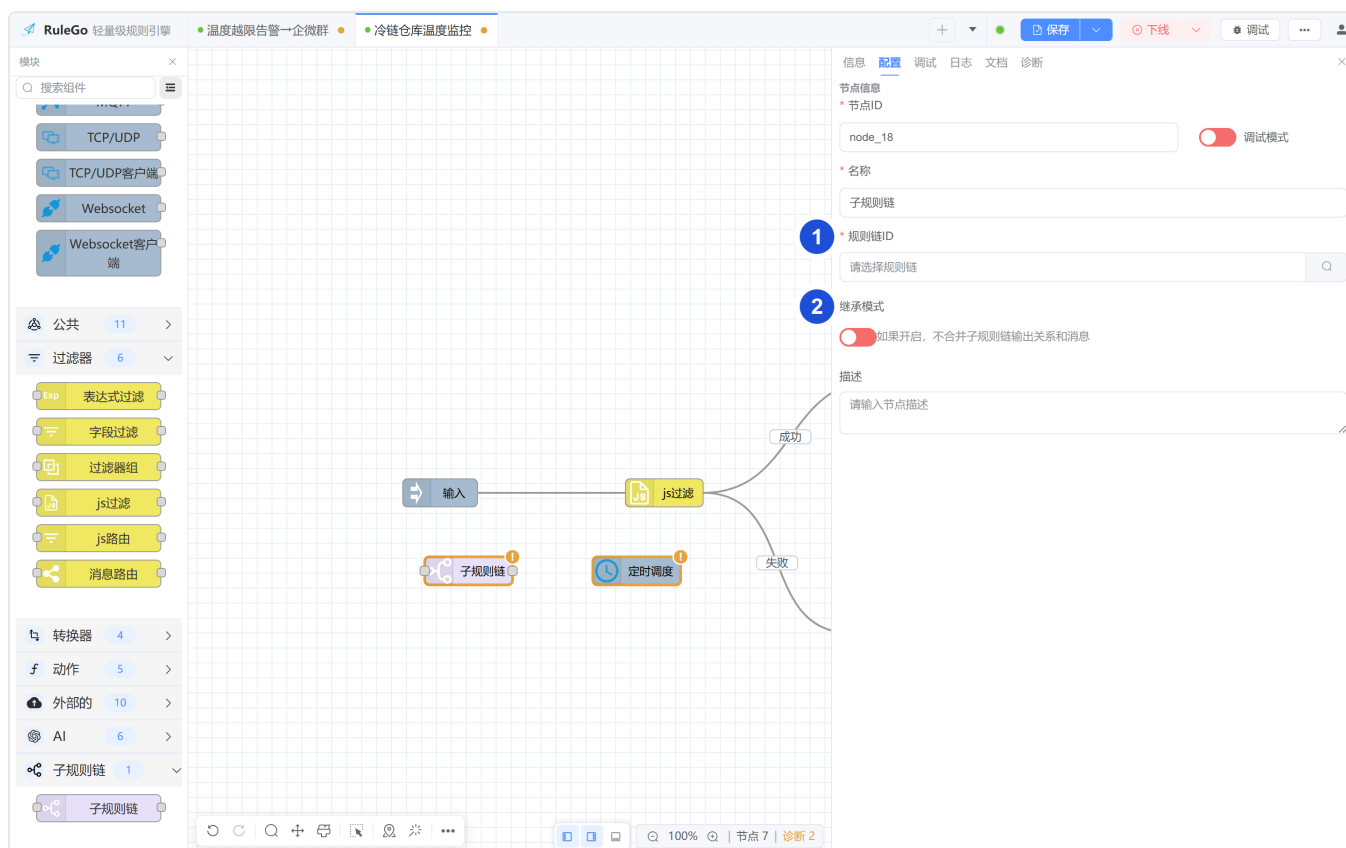


- ① **功能标签**：「已安装」列出当前组件及启停状态；「组件市场」「从市场安装」「从 DSL 安装」提供新组件来源；「国际化配置」可定制组件显示文案。

安装后的动态组件与内置组件一样出现在组件库与节点选择器中。

10.2 子规则链：把链当组件复用

一条规则链可以整体作为另一个链里的**节点**使用——这就是子规则链，规则链层的「组件化」。入口：组件库「**子规则链**」分组，把组件拖入画布：



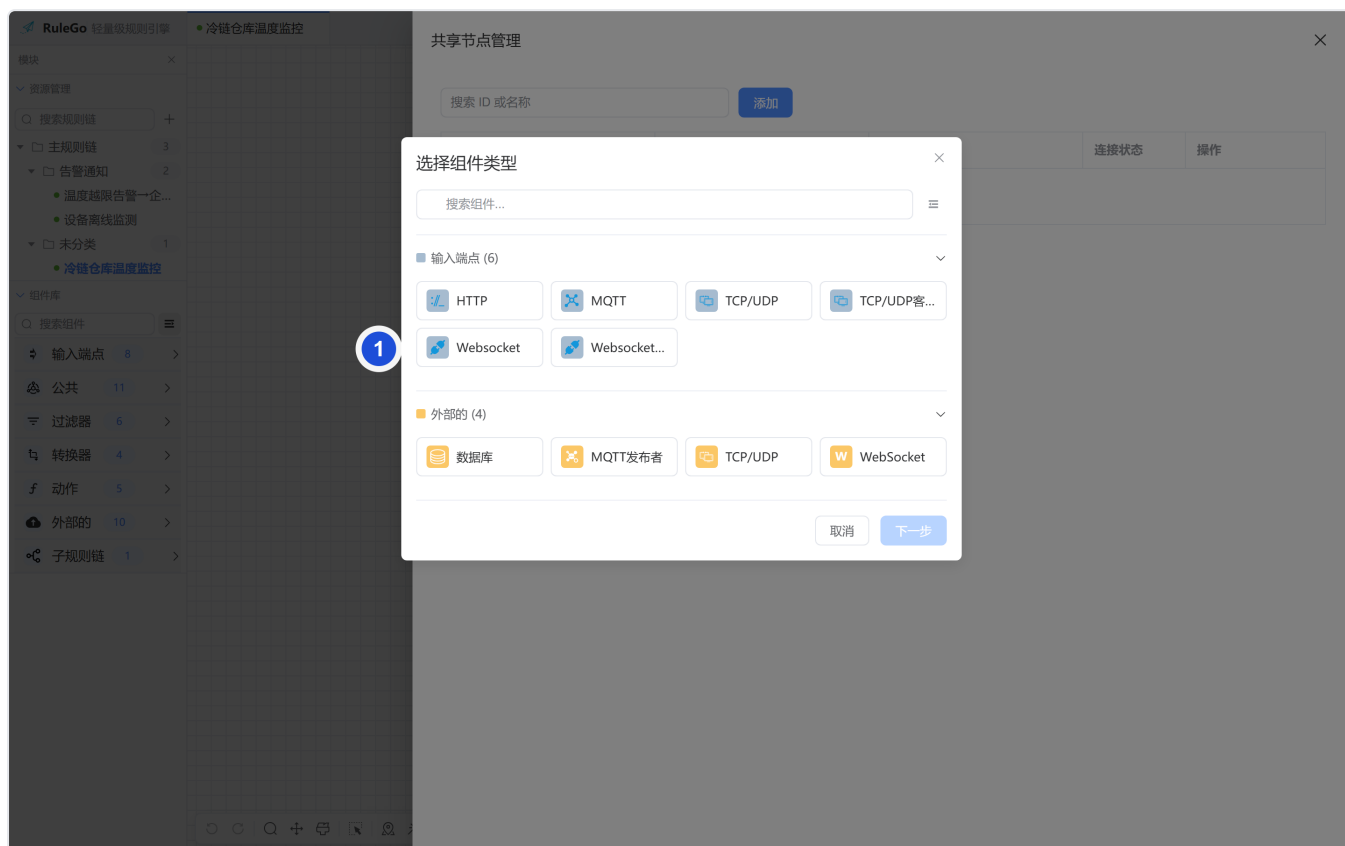
- ① **规则链ID**：搜索并选择要引用的目标链；消息到达该节点时转入目标链执行，执行完回到当前链继续流转。
- ② **继承模式**：开启后不合并子规则链的输出关系与消息，子链结果作为整体返回。

典型用法：把「告警通知」做成一条通用子链（组装文案、推送企微），各业务链判断出告警后统一调用它——通知逻辑改一处，所有业务链同步生效。

新建链时也可以直接声明为子规则链：规则链管理抽屉的新建入口选择「子规则链」类型。子链与主链一样可独立打开、调试、下线。

10.3 共享节点

数据库连接、MQTT 连接这类**被多条链共用**的连接配置，适合做成共享节点统一维护。入口：顶栏「…」更多操作菜单 → **规则链** → **共享节点**：



- ① 「添加共享节点」对话框：选择组件类型（MQTT、数据库客户端等）后进入下一步填写连接参数。

按向导选择组件类型（如 MQTT、数据库客户端），填写连接参数保存。共享节点会出现在资源管理面板的「共享节点」分组中，链内通过 `ref://` 引用，改一处全链生效。

11 规则链市场

11.1 浏览市场

官方维护了一批典型场景的规则链模板。入口：顶栏「…」更多操作菜单 → 规则链 → 规则链市场：

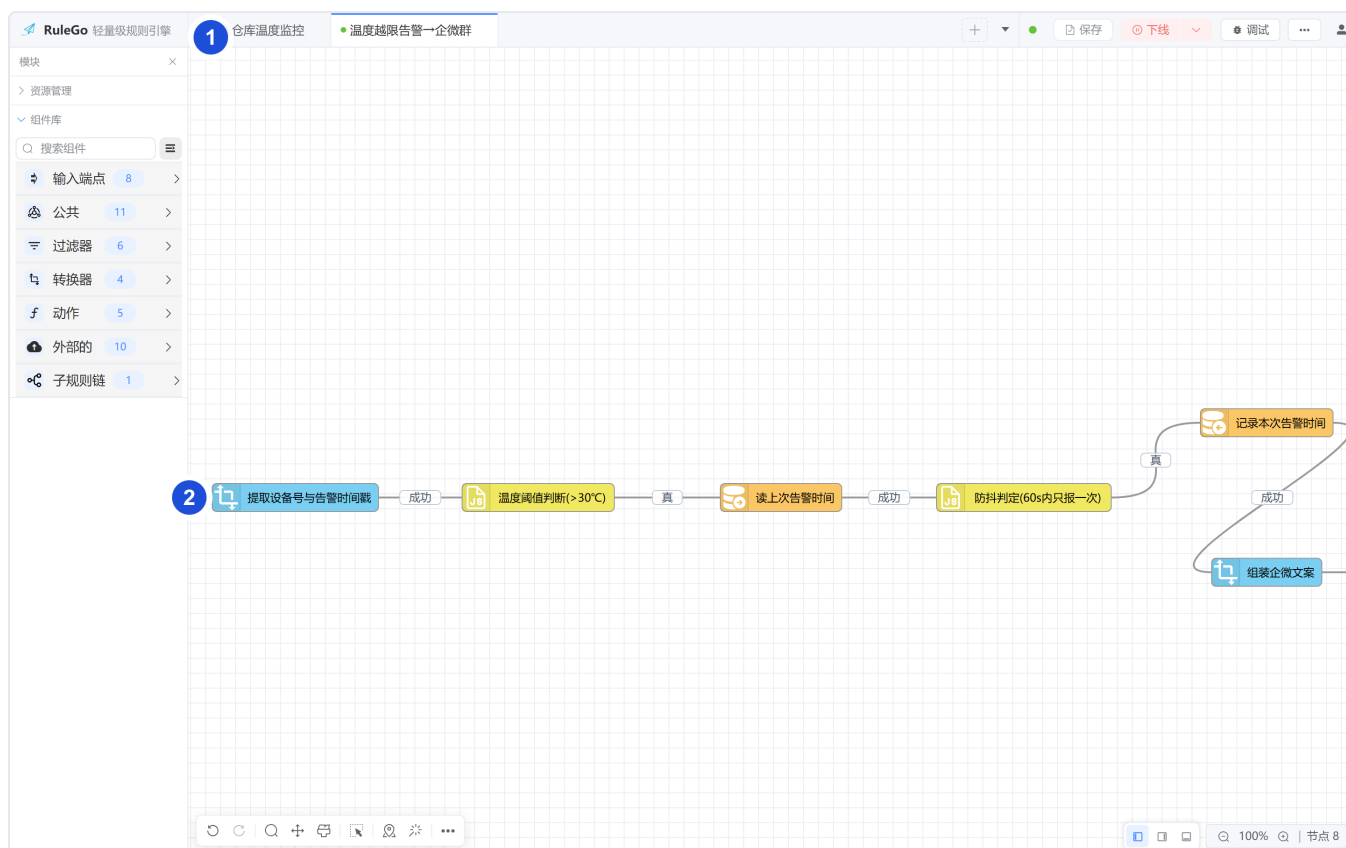


- ① **模板卡片**：显示模板名称、版本与功能说明，涵盖 MQTT 遥测入库、告警路由、Modbus 采集、AI 研判等常见场景。

顶部可在「规则链列表 / 子规则链列表」间切换。

11.2 导入模板

找到需要的模板，点击卡片上的「导入」并确认，模板链即安装到本地。以「温度越限告警→企微信群」为例——正好补充冷链监控的告警推送环节：



- ① **新链标签页**：导入的模板自动打开，标签页即模板名。
- ② **模板链画布**：完整的节点编排（MQTT 订阅 → 阈值判断 → 防抖 → 文案组装 → 推送企微信群），可在其基础上改成自己的参数。

导入的链与自建链地位相同：出现在资源树、可编辑、可调试、可下线。「下载」按钮则仅导出 DSL 文件不入库。

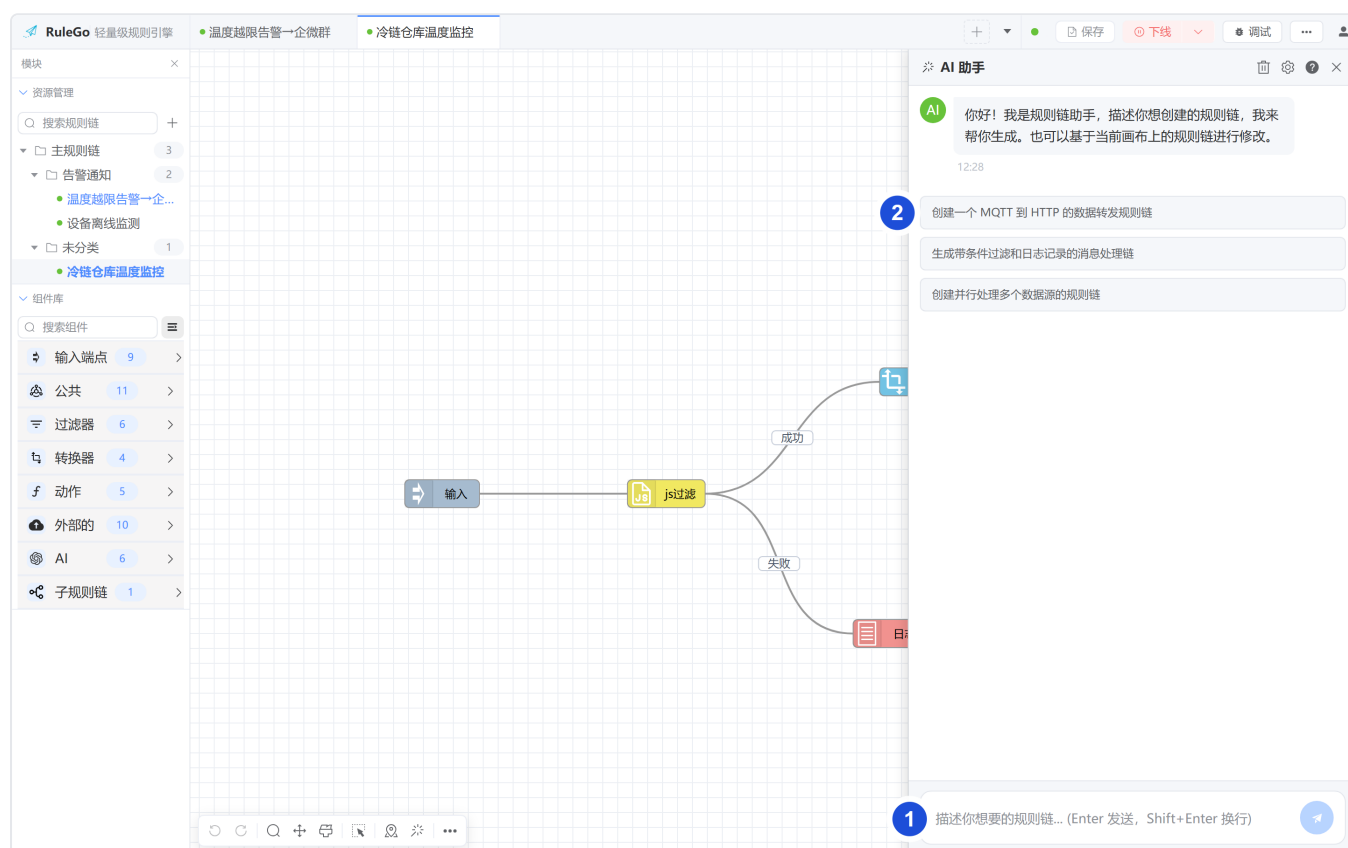
提示：模板链里的外部连接（MQTT broker、企微 webhook 等）需替换为自己的地址后才能实际运行。

12 AI 助手

编辑器内置 AI 助手：用自然语言描述需求即可生成规则链，也能就当前链答疑、给配置建议。本章介绍聊天面板、AI 生成规则链与相关设置。

12.1 打开聊天面板

空画布（新建链或草稿）时，点击引导卡里的「**AI 生成**」；已打开链时，在命令面板（**Ctrl+Shift+P**）执行「**AI 助手**」。面板在编辑区右侧展开：



- ① **输入框**：描述你想要的规则链或问题，Enter 发送、Shift+Enter 换行；按上下方向键可翻看输入历史。
- ② **快捷提示词**：点击直接填入，如「创建一个 MQTT 到 HTTP 的数据转发规则链」。

面板顶部的操作：**清空对话**（需确认）、**AI 助手设置**、**帮助文档**。助手回复中的思考过程与工具调用默认折叠，点击可展开查看；最后一条回复支持**重新生成**。

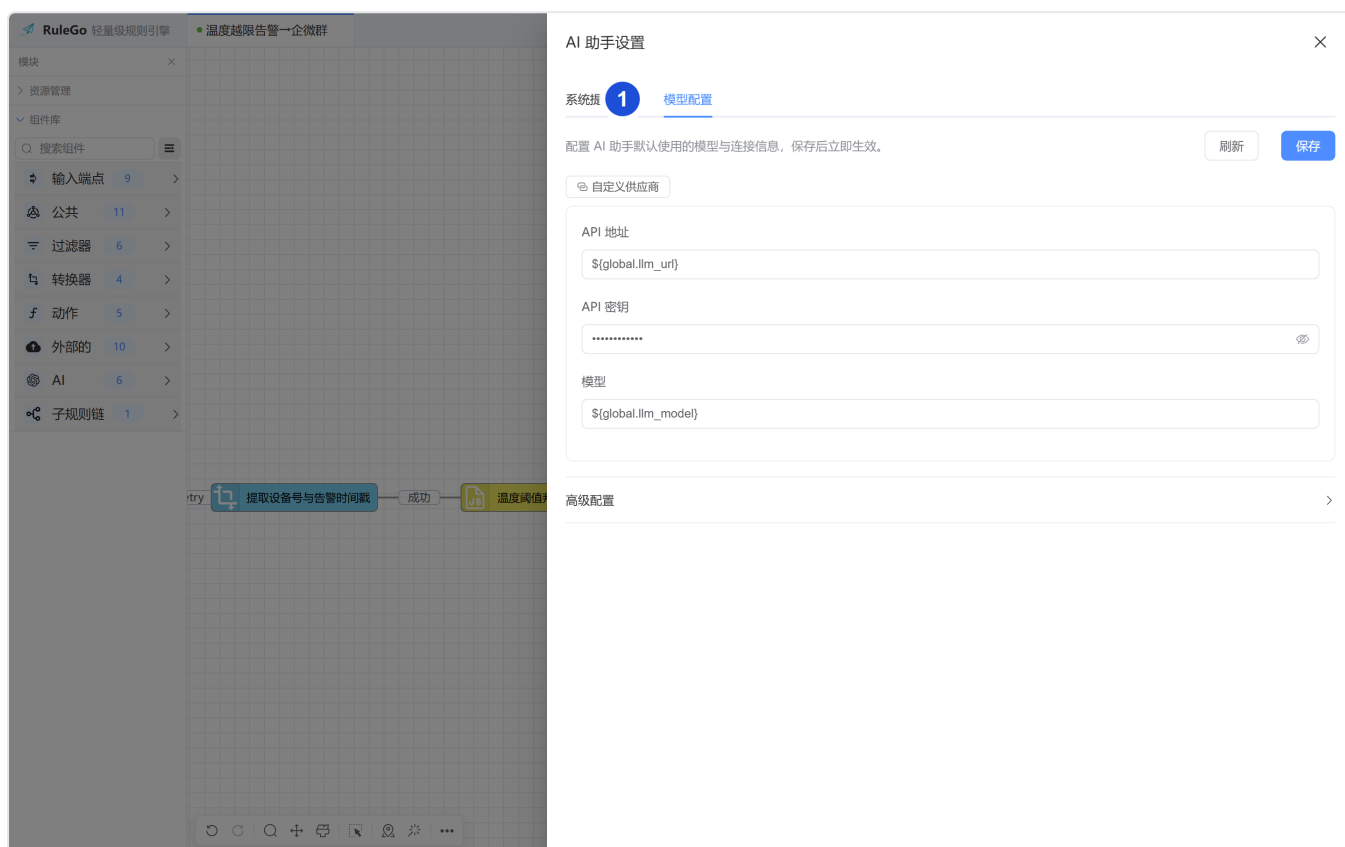
12.2 AI 生成规则链

在输入框描述需求，例如「创建一条链：MQTT 订阅 device/+/telemetry，温度超过 30 就记日志」。助手会给出规则链方案，并可直接应用到画布——空画布时自动载入，生成后可继续手工调整节点与连线，与自建链完全相同。

生成只是起点：AI 给的是可运行的参考实现，阈值、地址等参数仍需按实际业务核对修改。

12.3 AI 助手设置

点击用户胶囊 → **AI 助手设置**，两个标签分别管理提示词与模型：



- ① **模型配置**：填写模型服务商地址、API Key、模型名称与参数；保存后对话立即使用新配置。

「系统提示词」标签可编辑内置助手 `_assistant` 的 AGENTS.md——定义助手角色、能力与工作流程，懂 Prompt 的用户可以定制自己的助手行为。

提示：模型未配置时发送消息会收到引导提示；密钥保存在服务端，换浏览器无需重填。

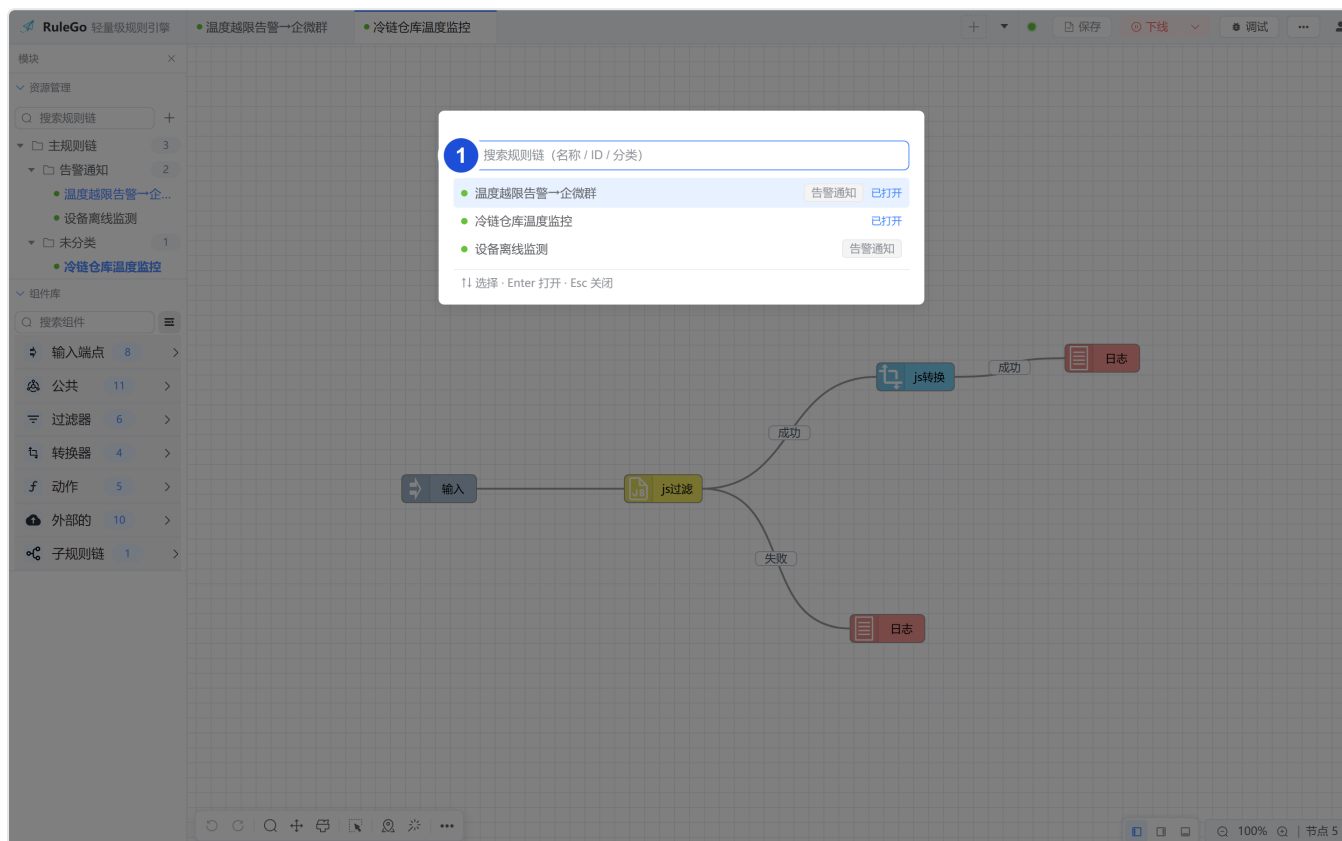
12.4 全局技能

用户菜单 → **全局技能**：管理助手可调用的技能（如查询设备、执行操作的定制能力）。技能由后端统一存储，对话中助手会按需自动调用；节点配置面板的「管理技能」入口可跳转同一管理界面。

13 效率工具与设置

13.1 快速切换规则链（Ctrl+P）

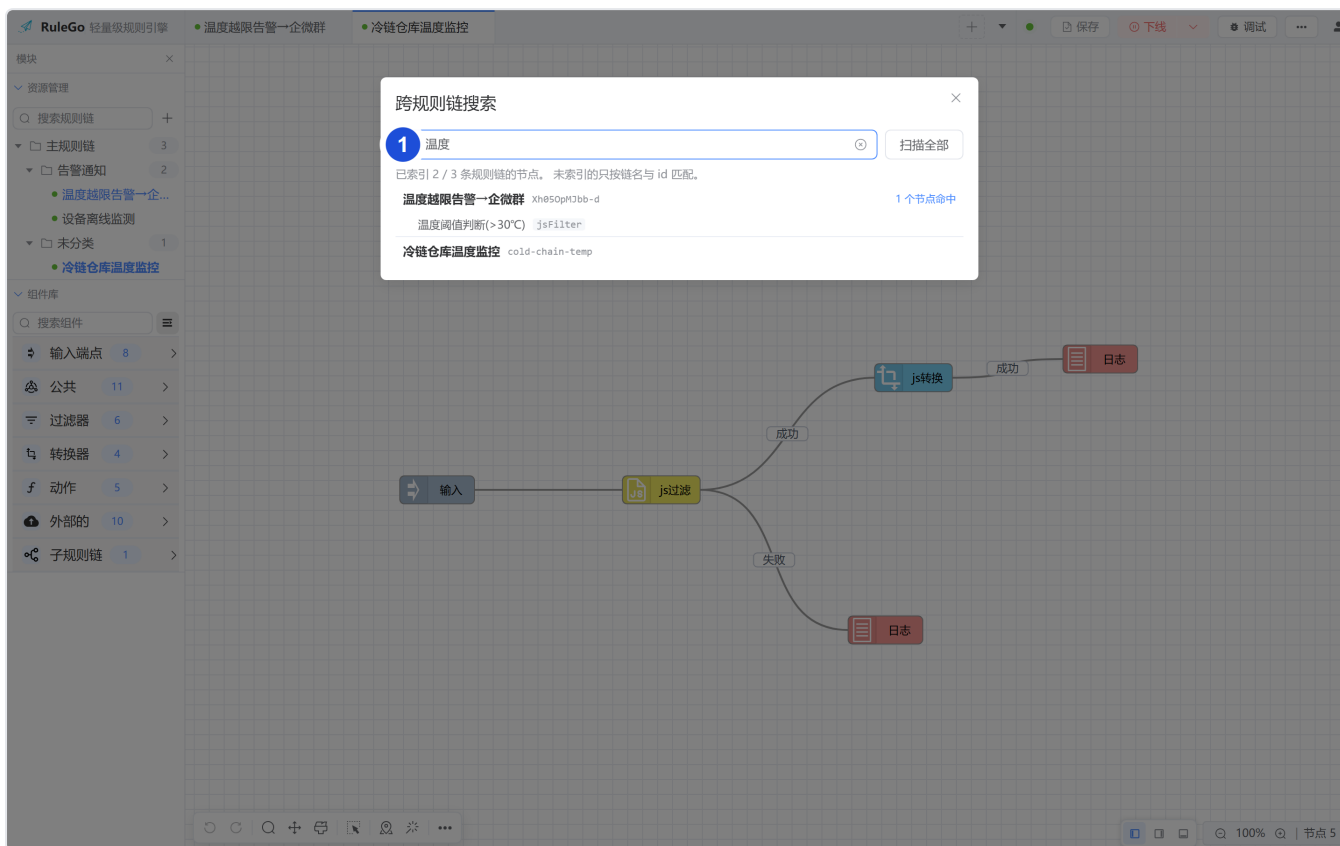
链多了以后，**Ctrl+P** 呼出快速切换器，输入关键字过滤后回车直接跳转：



- ① **搜索框**：按名称 / ID / 分类模糊匹配，列出候选链。

13.2 跨链全局搜索 (Ctrl+Shift+F)

在所有规则链里搜内容——找某个字段被哪些链引用时特别有用：

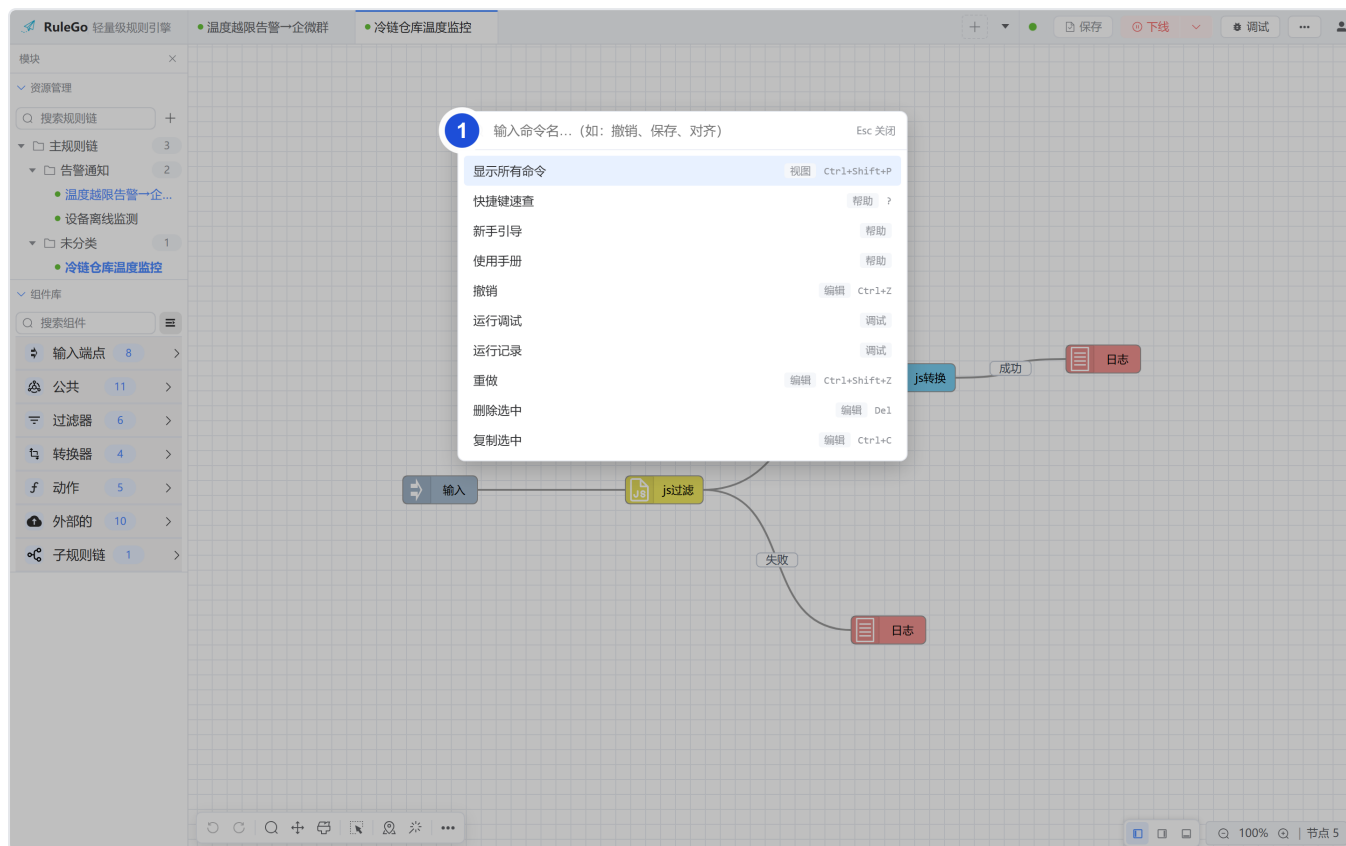


- ① **搜索输入**：输入关键字（如「温度」），结果按链分组列出命中的节点与配置，点击直达对应链。

首次使用时后台会逐步建立索引，未索引的链按名称与 ID 匹配。

13.3 命令面板 (Ctrl+Shift+P)

所有菜单动作都能从命令面板触达，键盘党的主入口：



- ① **命令输入**：输入动作名（如「导入」「组件管理」「新手引导」）回车执行。

13.4 快捷键一览

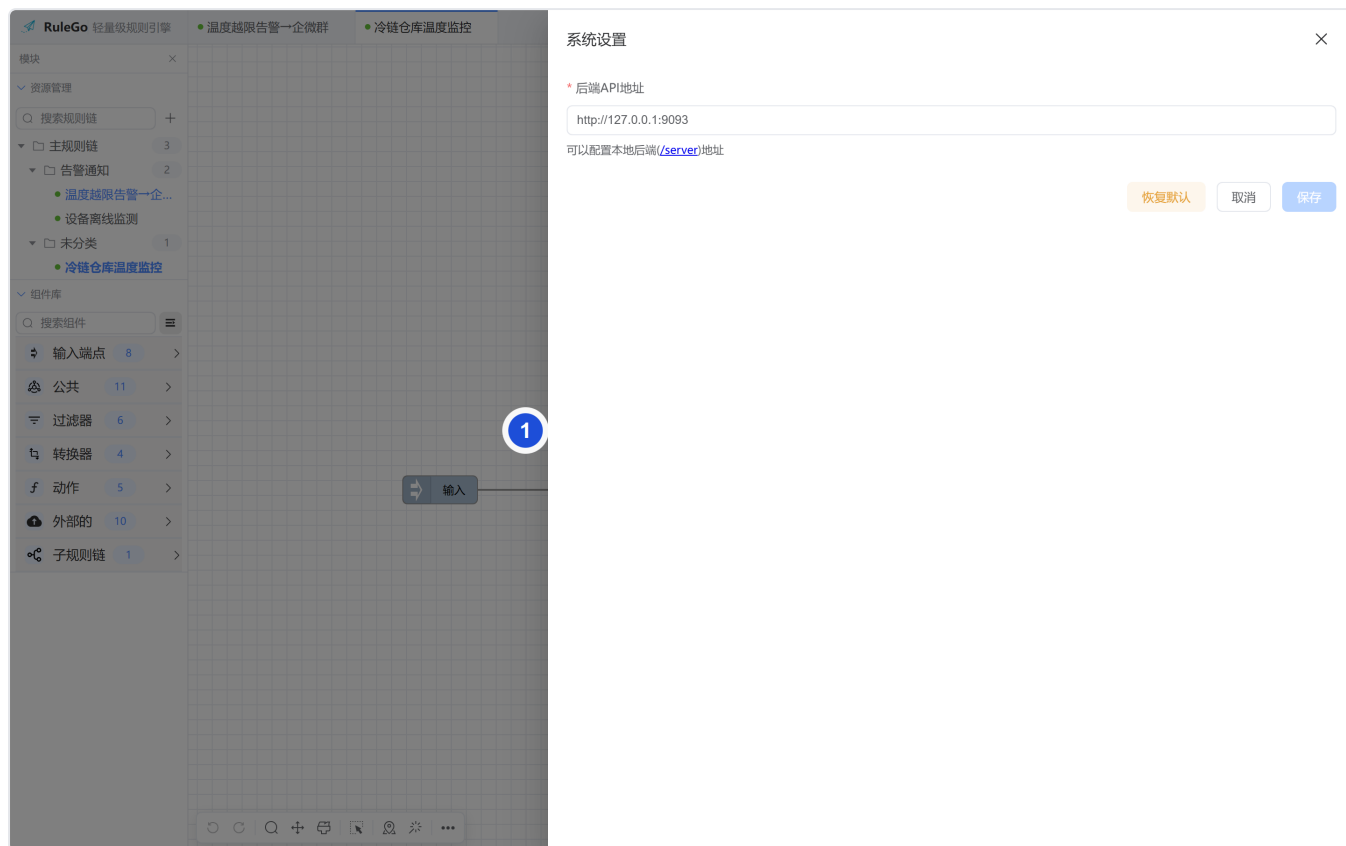
顶栏「…」更多操作菜单 → 外观与帮助 → 快捷键（或 `Ctrl+/,`）打开速查表：



- ① **快捷键面板**：分类列出画布、标签页、调试等全部快捷键。

13.5 系统设置

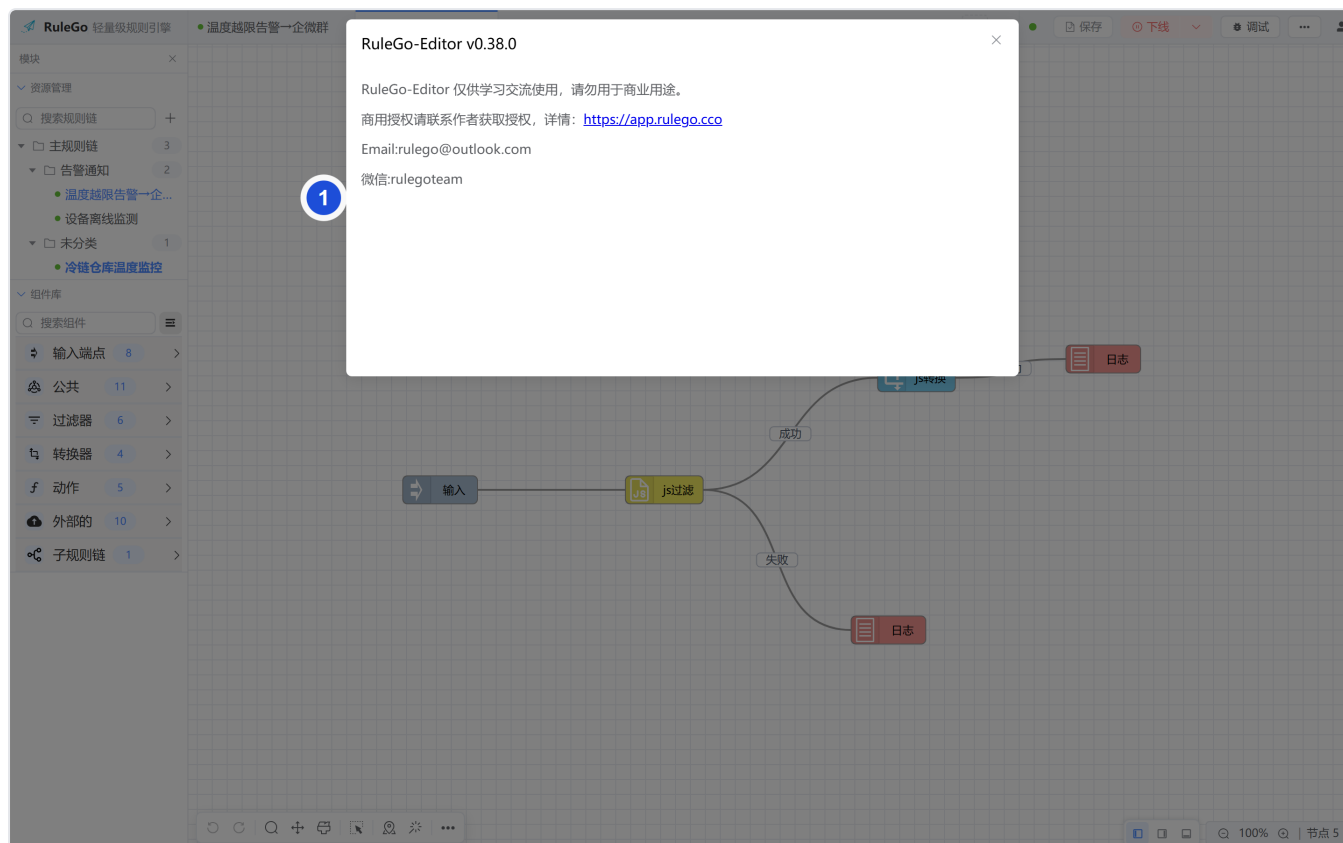
点击顶栏右侧用户胶囊 → **系统设置**：



- ① **系统设置抽屉**：可修改后端连接地址（连接状态灯也可直接点开）、语言等；连接地址改动后立即生效。服务端部署与配置见第 14 章。

13.6 关于

顶栏「…」更多操作菜单 → 外观与帮助 → 关于 查看版本信息：



- ① 关于对话框：显示产品名称与当前版本号。

常用快捷键速记：Ctrl+S 保存 · Ctrl+P 切链 · Ctrl+Shift+F 全局搜索 · Ctrl+Shift+P 命令面板 · Ctrl+/ 快捷键表 · Ctrl+B 组件库开关。

14 附录：服务端部署与配置

编辑器由 RuleGo Server 同源托管，本章面向负责部署的管理员，给出从下载到配置的最小路径。完整配置项参考官方文档站的《RuleGo-Server · 安装与部署》。

14.1 下载与启动

1. 从 GitHub / Gitee 的 RuleGo-Server Releases 下载对应平台的可执行文件（如 Windows 的 `rulego-server.exe`、Linux 的 `rulego-server`），或从源码编译。
2. 把可执行文件放到任意目录，直接运行：

```
./rulego-server      # Linux / macOS
rulego-server.exe    # Windows
```

3. 启动后默认监听 `:9090`，浏览器打开 `http://localhost:9090/editor/` 即是编辑器（本文档第 01 章的部署地址）。

服务无外部依赖（内嵌存储），数据落在 `data_dir` 指定的目录，备份该目录即备份全部规则链与配置。

14.2 配置文件

首次启动会在程序同目录生成 `config.conf`（INI 格式），常用配置项：

配置项	默认值	说明
<code>server</code>	<code>:9090</code>	监听地址与端口
<code>data_dir</code>	<code>./data</code>	数据目录（规则链、组件、运行日志等）
<code>require_auth</code>	<code>false</code>	是否开启登录鉴权
<code>default_username</code>	<code>admin</code>	未鉴权时使用的默认账号
<code>allow_cors</code>	<code>false</code>	编辑器与后端分离部署时需开启跨域
<code>marketplace_local_dir</code>	空	本地组件市场目录（见下）
<code>run_log_mode</code>	<code>off</code>	运行日志记录级别： <code>off</code> 不记录 / <code>summary</code> 摘要 / <code>detail</code> 含节点明细

配置项	默认值	说明
<code>skill_path</code>	<code>./skills</code>	AI 技能存储目录

配置示例：

```
server = :9090
data_dir = ./data
require_auth = true
marketplace_local_dir = /opt/rulego/marketplace
run_log_mode = summary

[users]
admin = admin
```

提示：修改配置后需重启服务生效。`config.conf` 不存在时全部使用默认值（鉴权关闭、监听 9090）。

14.3 账号与鉴权

- **开启鉴权：**`require_auth = true`。开启后编辑器出现登录页（第 01 章），REST API 需携带凭证。
- **账号管理：**`[users]` 段每行一个 `用户名 = 密码`；`admin` 为管理员，可进入编辑器的「租户管理」维护用户。
- **接口鉴权：**客户端程序推荐用 `X-API-Key` 请求头携带 API Key 调用 REST 接口（第 09 章）；浏览器交互走登录令牌（`Authorization: Bearer`）。
- **登录保护：**内置登录失败限流（默认 10 次/分钟），可用 `login_max_attempts`、`login_window_seconds` 调整。

14.4 组件市场

`marketplace_local_dir` 指向一个本地目录后，其中的规则链模板会出现在编辑器的「规则链市场」（第 11 章），供用户一键导入。目录结构就是一组规则链 DSL 文件，适合企业内部分发自己的模板库。

14.5 运行日志

`run_log_mode = summary`（或 `detail`）开启后，每次链执行自动留痕，编辑器底部「运行记录」面板（第 08 章）才有数据可看。`detail` 额外记录节点级明细，配合调试模式（第 07 章）定位问题；生产环境按需权衡存储开销。