|  |
| --- |
| 软件工程实验I组 |
| 测试报告v1.0.1 |
| 对Node-RED工具的消息聚集和图形可视化拓展 |

|  |
| --- |
| 暴明坤、夏欣怡、胡俊涛、张雨濛、叶柏威、刘子渊  2020-5-22 |

版本变更历史

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 版本 | 提交日期 | 编制人 | 修改说明 | 审核人 | 版本说明 |
| V1.0.1 | 2020/5/22 | 张雨濛 | 根据评审意见进行部分内容的修订 | 暴明坤 | 一稿 |
| V1.0.0 | 2020/5/19 | 张雨濛 | 初稿 | 暴明坤，夏欣怡，胡俊涛，张雨濛，叶柏威，刘子渊 |  |

目录

[1 引言 1](#_Toc41039090)

[1.1 编写目的 1](#_Toc41039091)

[1.2 测试范围 1](#_Toc41039092)

[2 测试总结 1](#_Toc41039093)

[2.1 测试用例执行结果 1](#_Toc41039094)

[2.2 覆盖率测试结果报告 8](#_Toc41039095)

[3 拓展功能测试报告 9](#_Toc41039096)

[3.1 创建飞书激活节点测试 9](#_Toc41039097)

[3.1.1 飞书激活节点测试 9](#_Toc41039098)

[3.2 创建飞书配置节点测试 11](#_Toc41039099)

[3.2.1 飞书配置节点功能测试 11](#_Toc41039100)

[3.3 创建飞书获取群列表节点测试 13](#_Toc41039101)

[3.3.1 飞书获取群列表节点加载测试 13](#_Toc41039102)

[3.3.2 飞书获取群列表节点功能测试 14](#_Toc41039103)

[3.4 创建飞书获得群信息节点测试 17](#_Toc41039104)

[3.4.1 飞书获得群信息节点功能测试 17](#_Toc41039105)

[3.5 创建飞书获得接收消息节点测试 25](#_Toc41039106)

[3.5.1 飞书获得接收消息节点接受私聊文本测试 25](#_Toc41039107)

[3.5.2 飞书获得接收消息节点接受私聊图片测试 28](#_Toc41039108)

[3.5.3 飞书获得接收消息节点接受群聊文本测试 31](#_Toc41039109)

[3.5.4 飞书获得接收消息节点接受群聊图片测试 34](#_Toc41039110)

[3.6 创建飞书获得上传图片节点测试 37](#_Toc41039111)

[3.6.1 飞书获得上传图片节点加载测试 37](#_Toc41039112)

[3.6.2 飞书获得上传图片节点功能测试 39](#_Toc41039113)

[3.7 创建飞书获取图片节点测试 41](#_Toc41039114)

[3.7.1 飞书获取图片节点加载测试 41](#_Toc41039115)

[3.7.2 飞书获取图片节点功能测试 43](#_Toc41039116)

[3.8 创建飞书添加目标节点测试 45](#_Toc41039117)

[3.8.1 飞书添加目标节点部署测试 45](#_Toc41039118)

[3.8.2 飞书添加目标节点正则表达式筛选器测试 46](#_Toc41039119)

[3.8.3 飞书添加目标节点列表筛选器测试 49](#_Toc41039120)

[3.8.4 飞书添加目标节点筛选对象及属性测试 52](#_Toc41039121)

[3.8.5 飞书添加目标节点直接添加模式测试 58](#_Toc41039122)

[3.8.6 飞书添加目标节点初始化测试 61](#_Toc41039123)

[3.8.7 飞书添加目标节点重复添加测试 64](#_Toc41039124)

[3.8.8 飞书添加目标节点异常处理测试 67](#_Toc41039125)

[3.8.9 飞书添加目标节点配置异常处理测试 76](#_Toc41039126)

[3.9 创建飞书移除目标节点测试 82](#_Toc41039127)

[3.9.1 飞书移除目标节点部署测试 82](#_Toc41039128)

[3.9.2 飞书移除目标节点正则表达式筛选器测试 83](#_Toc41039129)

[3.9.3 飞书移除目标节点列表筛选器测试 86](#_Toc41039130)

[3.9.4 飞书移除目标节点筛选对象及属性测试 89](#_Toc41039131)

[3.9.5 飞书移除目标节点初始化测试 95](#_Toc41039132)

[3.9.6 飞书移除目标节点异常处理测试 98](#_Toc41039133)

[3.9.7 飞书移除目标节点配置异常处理测试 102](#_Toc41039134)

[3.10 创建飞书发送消息节点测试 106](#_Toc41039135)

[3.10.1 飞书发送消息节点加载测试 106](#_Toc41039136)

[3.10.2 飞书发送消息节点发送群聊图片测试 108](#_Toc41039137)

[3.10.3 飞书发送消息节点发送群聊文本测试 111](#_Toc41039138)

[3.10.4 飞书发送消息节点发送私聊图片测试 113](#_Toc41039139)

[3.10.5 飞书发送消息节点发送私聊文本测试 116](#_Toc41039140)

[3.11 创建图表渲染节点测试 119](#_Toc41039141)

[3.11.1 图表渲染节点部署测试 119](#_Toc41039142)

[3.11.2 图表渲染节点Line chart绘制测试 120](#_Toc41039143)

[3.11.3 图表渲染节点Bar chart绘制测试 126](#_Toc41039144)

[3.11.4 图表渲染节点Pie chart绘制测试 128](#_Toc41039145)

[3.11.5 图表渲染节点Radar chart绘制测试 130](#_Toc41039146)

[3.11.6 图表渲染节点Polar area chart绘制测试 132](#_Toc41039147)

[3.11.7 图表渲染节点修改图表长宽测试 135](#_Toc41039148)

[3.11.8 图表渲染节点数据窗口功能测试 136](#_Toc41039149)

[3.11.9 图表渲染节点binary输出测试 138](#_Toc41039150)

[3.12 创建模板渲染节点测试 140](#_Toc41039151)

[3.12.1 模板渲染节点部署测试 140](#_Toc41039152)

[3.12.2 模板渲染节点功能正确性测试 141](#_Toc41039153)

[3.12.3 模板渲染节点异常处理测试 143](#_Toc41039154)

[3.12.4 模板渲染节点特殊输入处理测试 144](#_Toc41039155)

[3.12.5 模板渲染节点输出时清空字段功能正确性测试 146](#_Toc41039156)

[3.12.6 模板渲染节点输出后不清空字段功能正确性测试 149](#_Toc41039157)

[3.12.7 模板渲染节点解析模板时忽略无效占位符功能正确性测试 151](#_Toc41039158)

[3.12.8 模板渲染节点不等待所有字段功能正确性测试 153](#_Toc41039159)

[3.13 创建窗口节点测试 155](#_Toc41039160)

[3.13.1 窗口节点部署测试 155](#_Toc41039161)

[3.13.2 窗口节点配置异常处理测试 156](#_Toc41039162)

[3.13.3 窗口节点session模式功能正确性测试 157](#_Toc41039163)

[3.13.4 窗口节点session模式首次输入激活测试 160](#_Toc41039164)

[3.13.5 窗口节点session模式配置异常处理测试 162](#_Toc41039165)

[3.13.6 窗口节点tumbling时间模式功能正确性测试 163](#_Toc41039166)

[3.13.7 窗口节点tumbling时间模式无数据测试 166](#_Toc41039167)

[3.13.8 窗口节点tumbling时间模式配置异常处理测试 168](#_Toc41039168)

[3.13.9 窗口节点tumbling数据模式功能正确性测试 169](#_Toc41039169)

[3.13.10 窗口节点tumbling数量模式配置异常处理测试 172](#_Toc41039170)

[3.13.11 窗口节点sliding时间模式功能正确性测试 173](#_Toc41039171)

[3.13.12 窗口节点sliding时间模式窗口时长配置异常处理测试 176](#_Toc41039172)

[3.13.13 窗口节点sliding时间模式滑动时长配置异常处理测试 177](#_Toc41039173)

[3.13.14 窗口节点sliding数量模式功能正确性测试 178](#_Toc41039174)

[3.13.15 窗口节点sliding数量模式窗口数量配置异常处理测试 181](#_Toc41039175)

[3.13.16 窗口节点sliding数量模式滑动数量配置异常处理测试 182](#_Toc41039176)

[4 示例场景测试报告 183](#_Toc41039177)

[4.1 飞书机器人基本功能测试 183](#_Toc41039178)

[4.1.1 获取当前CPU使用情况图测试 183](#_Toc41039179)

[4.1.2 获取CPU使用情况订阅测试 184](#_Toc41039180)

[4.1.3 关闭CPU使用情况订阅测试 186](#_Toc41039181)

[4.2 飞书机器人乱序测试 187](#_Toc41039182)

[4.2.1 未订阅时申请关闭CPU使用情况订阅测试 187](#_Toc41039183)

[4.2.2 订阅中获取CPU使用情况测试 188](#_Toc41039184)

[4.3 飞书机器人错误输入测试 190](#_Toc41039185)

[4.3.1 向飞书机器人发送非预定格式的请求测试 190](#_Toc41039186)

[5 非功能需求测试报告 192](#_Toc41039187)

[5.1 负载/压力测试 192](#_Toc41039188)

[5.2 效能测试 192](#_Toc41039189)

[5.3 软件辅助功能测试 192](#_Toc41039190)

[5.4 全球化测试 192](#_Toc41039191)

[5.5 兼容性测试 192](#_Toc41039192)

[6 综合评价 193](#_Toc41039193)

# 引言

## 编写目的

**本测试报告是**针对**软件工程实验I组开发产品——“对Node-RED工具的消息聚集和图形可视化拓展”的测试报告，目的在于总结测试阶段的测试情况并分析测试结果，描述产品是否符合用户需求，是否已达到预期的功能目标，并对测试质量进行分析。**

## 测试范围

**测试主要依据《需求规格说明书v2.6.1》和《测试需求规格说明书v1.4.2》中的用例进行测试。**

# 测试总结

## 测试用例执行结果

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **测试用例标识** | **测试用例名称** | **状态** | **测试结果** | **所属测试需求** |
| **拓展功能测试** | | | | |
| TC001 | 飞书激活节点测试 | 已执行 | 测试通过 | 创建飞书激活节点测试 |
| TC002 | feishu-config节点功能测试 | 已执行 | 测试通过 | 创建飞书配置节点测试 |
| TC003 | feishu-all-group节点加载测试 | 已执行 | 测试通过 | 创建飞书获取群列表节点测试 |
| TC004 | feishu-all-group节点功能测试 | 已执行 | 测试通过 |
| TC005 | feishu-fetch-group-info节点功能测试 | 已执行 | 测试通过 | 创建飞书获得群信息节点测试 |
| TC006 | feishu-msg节点接收私聊文本测试 | 已执行 | 测试通过 | 创建飞书获得接收消息节点测试 |
| TC007 | feishu-msg节点接收私聊图片测试 | 已执行 | 测试通过 |
| TC008 | feishu-msg节点接收群聊文本测试 | 已执行 | 测试通过 |
| TC009 | feishu-msg节点接收群聊图片测试 | 已执行 | 测试通过 |
| TC010 | feishu-upload-image节点加载测试 | 已执行 | 测试通过 | 创建飞书获得上传图片节点测试 |
| TC011 | feishu-upload-image节点上传图片测试 | 已执行 | 测试通过 |
| TC012 | feishu-fetch-image节点加载测试 | 已执行 | 测试通过 | 创建飞书获取图片节点测试 |
| TC013 | feishu-fetch-image节点获取图片测试 | 已执行 | 测试通过 |
| TC014 | 飞书添加目标节点部署测试 | 已执行 | 测试通过 | 创建飞书添加目标节点测试 |
| TC015 | 飞书添加目标节点正则表达式筛选器测试 | 已执行 | 测试通过 |
| TC016 | 飞书添加目标节点列表筛选器测试 | 已执行 | 测试通过 |
| TC017 | 飞书添加目标节点筛选对象及属性测试 | 已执行 | 测试通过 |
| TC018 | 飞书添加目标节点直接添加模式测试 | 已执行 | 测试通过 |
| TC019 | 飞书添加目标节点初始化测试 | 已执行 | 测试通过 |
| TC020 | 飞书添加目标节点重复添加测试 | 已执行 | 测试通过 |
| TC021 | 飞书添加目标节点异常测试 | 已执行 | 测试通过 |
| TC022 | 飞书添加目标节点配置异常测试 | 已执行 | 测试通过 |
| TC023 | 飞书移除目标节点部署测试 | 已执行 | 测试通过 | 创建飞书移除目标节点测试 |
| TC024 | 飞书移除目标节点正则表达式筛选器测试 | 已执行 | 测试通过 |
| TC025 | 飞书移除目标节点列表筛选器测试 | 已执行 | 测试通过 |
| TC026 | 飞书移除目标节点筛选对象及属性测试 | 已执行 | 测试通过 |
| TC027 | 飞书移除目标节点初始化测试 | 已执行 | 测试通过 |
| TC028 | 飞书移除目标节点异常测试 | 已执行 | 测试通过 |
| TC029 | 飞书移除目标节点配置异常测试 | 已执行 | 测试通过 |
| TC030 | feishu-send节点加载测试 | 已执行 | 测试通过 | 创建飞书发送消息节点测试 |
| TC031 | feishu-send节点发送群聊图片测试 | 已执行 | 测试通过 |
| TC032 | feishu-send节点发送群聊文本测试 | 已执行 | 测试通过 |
| TC033 | feishu-send节点发送私聊图片测试 | 已执行 | 测试通过 |
| TC034 | feishu-send节点发送私聊文本测试 | 已执行 | 测试通过 |
| TC035 | chart-render节点部署测试 | 已执行 | 测试通过 | 创建图表渲染节点测试 |
| TC036 | chart-render节点Line chart绘制测试 | 已执行 | 测试通过 |
| TC037 | chart-render节点Bar chart绘制测试 | 已执行 | 测试通过 |
| TC038 | chart-render节点Pie chart绘制测试 | 已执行 | 测试通过 |
| TC039 | chart-render节点Radar chart绘制测试 | 已执行 | 测试通过 |
| TC040 | chart-render节点Polar area chart绘制测试 | 已执行 | 测试通过 |
| TC041 | chart-render节点修改图表长宽测试 | 已执行 | 测试通过 |
| TC042 | chart-render节点数据窗口测试 | 已执行 | 测试通过 |
| TC043 | chart-render节点binary输出测试 | 已执行 | 测试通过 |
| TC044 | 模板渲染节点部署测试 | 已执行 | 测试通过 | 创建模板渲染节点测试 |
| TC045 | 模板渲染节点功能正确性测试 | 已执行 | 测试通过 |
| TC046 | 模板渲染节点异常处理测试 | 已执行 | 测试通过 |
| TC047 | 模板渲染节点忽略空值输入测试 | 已执行 | 测试通过 |
| TC048 | 模板渲染节点输出时清空字段功能正确性测试 | 已执行 | 测试通过 |
| TC049 | 模板渲染节点输出后不清空字段功能正确性测试 | 已执行 | 测试通过 |
| TC050 | 模板渲染节点解析模板时忽略无效占位符功能正确性测试 | 已执行 | 测试通过 |
| TC051 | 模板渲染节点不等待所有字段功能正确性测试 | 已执行 | 测试通过 |
| TC052 | 窗口节点部署测试 | 已执行 | 测试通过 | 创建窗口节点测试 |
| TC053 | 窗口节点配置异常处理测试 | 已执行 | 测试通过 |
| TC054 | 窗口节点session模式功能正确性测试 | 已执行 | 测试通过 |
| TC055 | 窗口节点session模式首次输入激活测试 | 已执行 | 测试通过 |
| TC056 | 窗口节点session模式配置异常处理测试 | 已执行 | 测试通过 |
| TC057 | 窗口节点tumbling时间模式功能正确性测试 | 已执行 | 测试通过 |
| TC058 | 窗口节点tumbling时间模式无数据测试 | 已执行 | 测试通过 |
| TC059 | 窗口节点tumbling时间模式配置异常处理测试 | 已执行 | 测试通过 |
| TC060 | 窗口节点tumbling数据模式功能正确性测试 | 已执行 | 测试通过 |
| TC061 | 窗口节点tumbling数量模式配置异常处理测试 | 已执行 | 测试通过 |
| TC062 | 窗口节点sliding时间模式功能正确性测试 | 已执行 | 测试通过 |
| TC063 | 窗口节点sliding时间模式窗口时长配置异常处理测试 | 已执行 | 测试通过 |
| TC064 | 窗口节点sliding时间模式滑动时长配置异常处理测试 | 已执行 | 测试通过 |
| TC065 | 窗口节点sliding数量模式功能正确性测试 | 已执行 | 测试通过 |
| TC066 | 窗口节点sliding数量模式窗口数量配置异常处理测试 | 已执行 | 测试通过 |
| TC067 | 窗口节点sliding数量模式滑动数量配置异常处理测试 | 已执行 | 测试通过 |
| **示例场景测试** | | | | |
| TC068 | 获取当前CPU使用情况图测试 | 已执行 | 测试通过 |  |
| TC069 | 获取CPU使用情况订阅测试 | 已执行 | 测试通过 |  |
| TC070 | 关闭CPU使用情况订阅测试 | 已执行 | 测试通过 |  |
| TC071 | 未订阅时申请关闭CPU使用情况订阅测试 | 已执行 | 测试通过 |  |
| TC072 | 订阅中获取CPU使用情况测试 | 已执行 | 测试通过 |  |
| TC073 | 向飞书机器人发送非预定格式的请求测试 | 已执行 | 测试通过 |  |

## 覆盖率测试结果报告

**各模块覆盖结果如图1-3所示。**

图片包含 游戏机, 截图

描述已自动生成

图1 消息聚集部分节点测试覆盖率结果(1)

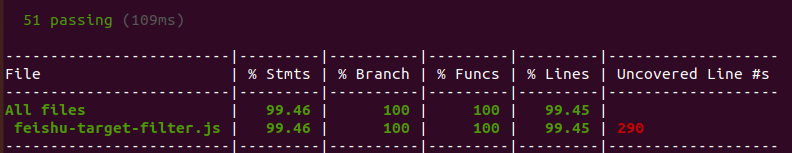


图2 消息聚集部分节点测试覆盖率结果(2)



图3 图表渲染部分节点测试覆盖率结果

# 拓展功能测试报告

## 创建飞书激活节点测试

* + 1. 飞书激活节点测试

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **测试用例名称** | | 飞书激活节点测试 | **测试用例标识** | | TC001 | | **测试需求标识** | | TR2.1 | |
| **简要描述** | | 测试飞书激活节点是否能正常工作 | | | | | | | | |
| **前提和约束** | | 无 | | | | | | | | |
| **测试方法** | | 单元测试 | | | | | | | | |
| **测试过程描述** | | | | | | | | | | |
| **序号** | **测试步骤** | | **输入** | **测试结果** | | | | **评价准则** | | **测试结论** |
| **预期结果** | | **实际结果** | |
| 1 | 搭建node-red测试环境 | |  |  | |  | |  | |  |
| 2 | 创建含有一个飞书激活节点的工作流，设置激活URL为/activate,  激活端口是60000 | |  |  | |  | |  | |  |
| 3 | 部署工作流 | |  |  | |  | |  | |  |
| 4 | 向测试端口发送激活验证消息 | | 激活消息  { challenge: 'fjaioefn afej@ 1239', token: 'fajiofjo 09j 90 n014', type: "url\_verification"  } | 返回响应200 | | 返回响应200 | | 实际结果与预期结果一致 | | 通过 |
| 5 | 停止node-red测试环境 | |  |  | |  | |  | |  |
| **备注** | | 预期返回响应  {  challenge: 'fjaioefn afej@ 1239'  } | | | | | | | | |

## 创建飞书配置节点测试

* + 1. 飞书配置节点功能测试

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **测试用例名称** | | feishu-config节点功能测试 | **测试用例标识** | | TC002 | | **测试需求标识** | | TR2.2 | |
| **简要描述** | | 测试 feishu-config节点是否能正确获得tenant-token | | | | | | | | |
| **前提和约束** | | 无 | | | | | | | | |
| **测试方法** | | 单元测试 | | | | | | | | |
| **测试过程描述** | | | | | | | | | | |
| **序号** | **测试步骤** | | **输入** | **测试结果** | | | | **评价准则** | | **测试结论** |
| **预期结果** | | **实际结果** | |
| 1 | 搭建node-red测试环境 | |  |  | |  | |  | |  |
| 2 | 创建含有一个feishu-config节点的工作流 | |  |  | |  | |  | |  |
| 3 | 按照飞书tenant\_access\_token API 架设nock拦截器， | |  |  | |  | |  | |  |
| 4 | 部署工作流 | |  |  | |  | |  | |  |
| 5 | 检查节点加载状态 | |  | feishu-config节点获得tenant\_token | | feishu-config节点获得tenant\_token | | 检查tenant\_token是否与预设值相同 | | 通过 |
| 6 | 停止node-red测试环境 | |  |  | |  | |  | |  |
| **备注** | | 无 | | | | | | | | |

## 创建飞书获取群列表节点测试

* + 1. 飞书获取群列表节点加载测试

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **测试用例名称** | | feishu-all-group  节点加载测试 | **测试用例标识** | | TC003 | | **测试需求标识** | | TR2.3 | |
| **简要描述** | | 测试 feishu-all-group节点是否能正确加载 | | | | | | | | |
| **前提和约束** | | 无 | | | | | | | | |
| **测试方法** | | 冒烟测试 | | | | | | | | |
| **测试过程描述** | | | | | | | | | | |
| **序号** | **测试步骤** | | **输入** | **测试结果** | | | | **评价准则** | | **测试结论** |
| **预期结果** | | **实际结果** | |
| 1 | 搭建node-red测试环境 | |  |  | |  | |  | |  |
| 2 | 创建含有一个feishu-all-group节点和一个feishu-config节点的工作流 | |  |  | |  | |  | |  |
| 3 | 按照飞书tenant\_access\_token API 架设nock拦截器 | |  |  | |  | |  | |  |
| 4 | 部署工作流 | |  |  | |  | |  | |  |
| 5 | 检查节点加载状态 | |  | feishu-config节点获得tenant\_token | | feishu-config节点获得tenant\_token | | 检查tenant\_token是否与预设值相同 | | 通过 |
|  | feishu-all-group节点没有报错 | | feishu-all-group节点没有报错 | | 检查是否有未处理的异常 | | 通过 |
| 6 | 停止node-red测试环境 | |  |  | |  | |  | |  |
| **备注** | | 无 | | | | | | | | |

* + 1. 飞书获取群列表节点功能测试

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **测试用例名称** | | feishu-all-group  节点功能测试 | **测试用例标识** | | TC004 | | **测试需求标识** | | TR2.3 | |
| **简要描述** | | 测试 feishu-all-group节点是否能获取群列表 | | | | | | | | |
| **前提和约束** | | 无 | | | | | | | | |
| **测试方法** | | 单元测试 | | | | | | | | |
| **测试过程描述** | | | | | | | | | | |
| **序号** | **测试步骤** | | **输入** | **测试结果** | | | | **评价准则** | | **测试结论** |
| **预期结果** | | **实际结果** | |
| 1 | 搭建node-red测试环境 | |  |  | |  | |  | |  |
| 2 | 创建含有一个feishu-all-group节点，一个feishu-config节点和一个helper节点的工作流 | |  |  | |  | |  | |  |
| 3 | 按照飞书tenant\_access\_token API 架设nock拦截器 | |  |  | |  | |  | |  |
| 4 | 按照飞书  获取群列表API架设nock拦截器 | |  |  | |  | |  | |  |
| 5 | 设置helper节点的接受消息回调 | |  |  | |  | |  | |  |
| 6 | 部署工作流 | |  |  | |  | |  | |  |
| 7 | 检查节点工作状态 | |  | feishu-config节点获得tenant\_token | | feishu-config节点获得tenant\_token | | 检查tenant\_token是否与预设值相同 | | 通过 |
|  | feishu-all-group节点没有报错 | | feishu-all-group节点没有报错 | | 检查是否有未处理的异常 | | 通过 |
| 测试消息1 | feishu-all-group节点请求到测试用群列表，并添加在返回消息上 | | feishu-all-group节点请求到测试用群列表，并添加在返回消息上 | | 检查返回消息的群列表是否与测试群列表相同 | | 通过 |
| 8 | 停止node-red测试环境 | |  |  | |  | |  | |  |
| **备注** | | 测试消息1  {  payload: {  msg: {  text: 'hello'  }  },  feishu\_meta\_info: {  target: {  chat: [],  user: []  },  chat: [],  user: []  }  } | | | | | | | | |

## 创建飞书获得群信息节点测试

* + 1. 飞书获得群信息节点功能测试

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **测试用例名称** | | feishu-fetch-group-info  节点功能测试 | **测试用例标识** | | TC005 | | **测试需求标识** | | TR2.4 | |
| **简要描述** | | 测试 feishu-fetch-group-info节点是否能获取群列表 | | | | | | | | |
| **前提和约束** | | 无 | | | | | | | | |
| **测试方法** | | 单元测试 | | | | | | | | |
| **测试过程描述** | | | | | | | | | | |
| **序号** | **测试步骤** | | **输入** | **测试结果** | | | | **评价准则** | | **测试结论** |
| **预期结果** | | **实际结果** | |
| 1 | 搭建node-red测试环境 | |  |  | |  | |  | |  |
| 2 | 创建含有一个feishu-fetch-group-info节点，一个feishu-config节点和一个helper节点的工作流 | |  |  | |  | |  | |  |
| 3 | 按照飞书tenant\_access\_token API 架设nock拦截器 | |  |  | |  | |  | |  |
| 4 | 按照飞书  获取群信息API架设nock拦截器 | |  |  | |  | |  | |  |
| 5 | 设置helper节点的接受消息回调 | |  |  | |  | |  | |  |
| 6 | 部署工作流 | |  |  | |  | |  | |  |
| 7 | 检查节点工作状态 | |  | feishu-config节点获得tenant\_token | | feishu-config节点获得tenant\_token | | 检查tenant\_token是否与预设值相同 | | 通过 |
| 测试消息1 | feishu-fetch-group-info节点请求到测试用群列表，并添加在返回消息上 | | feishu-fetch-group-info节点请求到测试用群列表，并添加在返回消息上 | | 检查返回消息的群信息是否与测试群信息相同 | | 通过 |
| 8 | 停止node-red测试环境 | |  |  | |  | |  | |  |
| **备注** | | 测试消息1  {  payload: {  msg: {  text: 'hello'  }  },  feishu\_meta\_info: {  target: {  chat: [],  user: []  },  chat: TEST\_GROUP\_LIST,  user: []  }  }  预期返回消息1  {  payload: {  msg: {  text: 'hello'  }  },  feishu\_meta\_info: {  target: {  chat: [],  user: []  },  chat: TEST\_GROUP\_INFOS,  user: []  }  }  TEST\_GROUP\_LIST  ['hello', 'world', '\_10920\_239301\_KLSDJFwio@']  TEST\_GROUP\_INFO  [  {  "avatar": "http://p3.pstatp.com/origin/78c0000676df676a7f6e",  "chat\_id": TEST\_GROUP\_LIST[0],  "description": "group description",  "i18n\_names": {  "en\_us": "English",  "ja\_jp": "日本語",  "zh\_cn": "中文"  },  "members": [  {  "open\_id": "ou\_6edde2deccabb76c12b30f0345f19aa1",  "user\_id": "8e17d887"  },  {  "open\_id": "ou\_194911f90c43ec42d1ba0e93f22b8fb1",  "user\_id": "ca51d83b"  },  ],  "name": "test group 1",  "type": "group",  "owner\_open\_id": "ou\_6edde2deccabb76c12b30f0345f19aa1",  "owner\_user\_id": "8e17d887"  },  {  "avatar": "http://p3.pstatp.com/origin/oe 2391021 @000676df676a7f6e",  "chat\_id": TEST\_GROUP\_LIST[1],  "description": "grre dis 2 iojfioaw ioji1",  "i18n\_names": {  "en\_us": "English",  "ja\_jp": "日本語",  "zh\_cn": "中文"  },  "members": [  {  "open\_id": "ou\_194911f90c43ec42d1ba0e93f22b8fb1",  "user\_id": "ca51d83b"  },  {  "open\_id": "ou\_b6671e895da5ef8f84ec8831c26348bd",  "user\_id": "24e1cbff"  }  ],  "name": "test group 2 i",  "type": "group",  "owner\_open\_id": "ou\_i2w3139-1dl",  "owner\_user\_id": "a- 1- 1323"  },  {  "avatar": "http://p3.pstatp.com/origin/oe 2391021 @000676df676a7f6e",  "chat\_id": TEST\_GROUP\_LIST[2],  "description": "grre dis 2 iojfioaw ioji1",  "i18n\_names": {  "en\_us": "English",  "ja\_jp": "日本語",  "zh\_cn": "中文"  },  "members": [  {  "open\_id": "ou\_194911f90c43ec42d1ba0e93f22b8fb1",  "user\_id": "ca51d83b"  },  {  "open\_id": "ou\_b6671e895da5ef8f84ec8831c26348bd",  "user\_id": "24e1cbff"  }  ],  "name": "test group 2 i",  "type": "group",  "owner\_open\_id": "ou\_i2w3139-1dl",  "owner\_user\_id": "a- 1- 1323"  },  ] | | | | | | | | |

## 创建飞书获得接收消息节点测试

* + 1. 飞书获得接收消息节点接受私聊文本测试

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **测试用例名称** | | feishu-msg  节点接收私聊文本功能测试 | **测试用例标识** | | TC006 | | **测试需求标识** | | TR2.5 | |
| **简要描述** | | 测试 feishu-msg节点是否能正确收到私聊文本消息 | | | | | | | | |
| **前提和约束** | | 无 | | | | | | | | |
| **测试方法** | | 单元测试 | | | | | | | | |
| **测试过程描述** | | | | | | | | | | |
| **序号** | **测试步骤** | | **输入** | **测试结果** | | | | **评价准则** | | **测试结论** |
| **预期结果** | | **实际结果** | |
| 1 | 搭建node-red测试环境 | |  |  | |  | |  | |  |
| 2 | 创建含有一个feishu-msg，一个有helper节点的工作流 | |  |  | |  | |  | |  |
| 3 | 设置helper的消息回调函数 | |  |  | |  | |  | |  |
| 4 | 部署工作流 | |  |  | |  | |  | |  |
| 5 | 检查节点工作状态 | | 请求1 | feishu-msg节点返回预期消息1 | | feishu-msg节点返回预期消息1 | | 返回的消息与预期是否相等 | | 通过 |
| 6 | 停止node-red测试环境 | |  |  | |  | |  | |  |
| **备注** | | 请求1  {  "uuid": "41b5f371157e3d5341b38b20396e77e3",  "token": "2g7als3DgPW6Xp1xEpmcvgVhQG621bFY",//校验Token  "ts": "1550038209.428520", //时间戳  "type": "event\_callback",//事件回调此处固定为event\_callback  "event": {  "type": "message", // 事件类型  "app\_id": "cli\_xxx",  "tenant\_key": "xxx", //企业标识  "root\_id": "",  "parent\_id": "",  "open\_chat\_id": "oc\_5ce6d572455d361153b7cb51da133945",  "chat\_type": "private",//私聊private，群聊group  "msg\_type": "text", //消息类型  "open\_id": TEST\_open\_id,  "open\_message\_id": "om\_36686ee62209da697d8775375d0c8e88",  "is\_mention": false,  "text": TEST\_TEXT, // 消息文本  "text\_without\_at\_bot": TEST\_without\_at //消息内容，会过滤掉at你的机器人的内容  }  预期消息1  {  payload: {  msg: {  text: TEST\_without\_at  }  }  },  feishu\_meta\_info: {  target: {  chat: [],  user: [TEST\_open\_id]  },  chat: [],  user: [TEST\_open\_id]  }  } | | | | | | | | |

* + 1. 飞书获得接收消息节点接受私聊图片测试

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **测试用例名称** | | feishu-msg  节点接收私聊图片功能测试 | **测试用例标识** | | TC007 | | **测试需求标识** | | TR2.5 | |
| **简要描述** | | 测试 feishu-msg节点是否能正确收到私聊图片消息 | | | | | | | | |
| **前提和约束** | | 无 | | | | | | | | |
| **测试方法** | | 单元测试 | | | | | | | | |
| **测试过程描述** | | | | | | | | | | |
| **序号** | **测试步骤** | | **输入** | **测试结果** | | | | **评价准则** | | **测试结论** |
| **预期结果** | | **实际结果** | |
| 1 | 搭建node-red测试环境 | |  |  | |  | |  | |  |
| 2 | 创建含有一个feishu-msg，一个有helper节点的工作流 | |  |  | |  | |  | |  |
| 3 | 设置helper的消息回调函数 | |  |  | |  | |  | |  |
| 4 | 部署工作流 | |  |  | |  | |  | |  |
| 5 | 检查节点工作状态 | | 请求1 | feishu-msg节点返回预期消息1 | | feishu-msg节点返回预期消息1 | | 返回的消息与预期是否相等 | | 通过 |
| 6 | 停止node-red测试环境 | |  |  | |  | |  | |  |
| **备注** | | 请求1  {  "uuid": "c58e17e9e84824a48e51a562cf969fb3",  "token": "2g7als3DgPW6Xp1xEpmcvgVhQG621bFY",  "ts": "1550038110.478493",  "type": "event\_callback",  "event": {  "type": "message",  "app\_id": "cli\_xxx",  "tenant\_key": "xxx", //企业标识  "root\_id": "",  "parent\_id": "",  "open\_chat\_id": "oc\_5ce6d572455d361153b7cb51da133945",  "chat\_type": "private",  "msg\_type": "image", //图片消息  "image\_height": "300",  "image\_width": "300",  "open\_id": TEST\_open\_id,  "open\_message\_id": "om\_340057d660022bf141eb470859c6114c",  "is\_mention": false,  "image\_key": TEST\_IMAGE\_KEY, // image\_key，获取图片内容请查https://open.feishu.cn/document/ukTMukTMukTM/uYzN5QjL2cTO04iN3kDN  }  }  预期消息1  {  payload: {  msg: {  image\_key: TEST\_IMAGE\_KEY  },  },  feishu\_meta\_info: {  target: {  chat: [],  user: [TEST\_open\_id]  },  chat: [],  user: [TEST\_open\_id]  }  } | | | | | | | | |

* + 1. 飞书获得接收消息节点接受群聊文本测试

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **测试用例名称** | | feishu-msg  节点接收群聊文本功能测试 | **测试用例标识** | | TC008 | | **测试需求标识** | | TR2.5 | |
| **简要描述** | | 测试 feishu-msg节点是否能正确收到群聊文本消息 | | | | | | | | |
| **前提和约束** | | 无 | | | | | | | | |
| **测试方法** | | 单元测试 | | | | | | | | |
| **测试过程描述** | | | | | | | | | | |
| **序号** | **测试步骤** | | **输入** | **测试结果** | | | | **评价准则** | | **测试结论** |
| **预期结果** | | **实际结果** | |
| 1 | 搭建node-red测试环境 | |  |  | |  | |  | |  |
| 2 | 创建含有一个feishu-msg，一个有helper节点的工作流 | |  |  | |  | |  | |  |
| 3 | 设置helper的消息回调函数 | |  |  | |  | |  | |  |
| 4 | 部署工作流 | |  |  | |  | |  | |  |
| 5 | 检查节点工作状态 | | 请求1 | feishu-msg节点返回预期消息1 | | feishu-msg节点返回预期消息1 | | 返回的消息与预期是否相等 | | 通过 |
| 6 | 停止node-red测试环境 | |  |  | |  | |  | |  |
| **备注** | | 请求1  {  "uuid": "41b5f371157e3d5341b38b20396e77e3",  "token": "2g7als3DgPW6Xp1xEpmcvgVhQG621bFY",//校验Token  "ts": "1550038209.428520", //时间戳  "type": "event\_callback",//事件回调此处固定为event\_callback  "event": {  "type": "message", // 事件类型  "app\_id": "cli\_xxx",  "tenant\_key": "xxx", //企业标识  "root\_id": "",  "parent\_id": "",  "open\_chat\_id": TEST\_open\_chat\_id,  "chat\_type": "group",//私聊private，群聊group  "msg\_type": "text", //消息类型  "open\_id": TEST\_open\_id,  "open\_message\_id": "om\_36686ee62209da697d8775375d0c8e88",  "is\_mention": false,  "text": TEST\_TEXT, // 消息文本  "text\_without\_at\_bot": TEST\_without\_at //消息内容，会过滤掉at你的机器人的内容  }  }  预期消息1  {  payload: {  msg: {  text: TEST\_without\_at  }  }  },  feishu\_meta\_info: {  target: {  chat: [TEST\_open\_chat\_id],  user: []  },  chat: [TEST\_open\_chat\_id],  user: []  }  } | | | | | | | | |

* + 1. 飞书获得接收消息节点接受群聊图片测试

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **测试用例名称** | | feishu-msg  节点接收群聊图片 | **测试用例标识** | | TC009 | | **测试需求标识** | | TR2.5 | |
| **简要描述** | | 测试 feishu-msg节点是否能正确收到群聊图片消息 | | | | | | | | |
| **前提和约束** | | 无 | | | | | | | | |
| **测试方法** | | 单元测试 | | | | | | | | |
| **测试过程描述** | | | | | | | | | | |
| **序号** | **测试步骤** | | **输入** | **测试结果** | | | | **评价准则** | | **测试结论** |
| **预期结果** | | **实际结果** | |
| 1 | 搭建node-red测试环境 | |  |  | |  | |  | |  |
| 2 | 创建含有一个feishu-msg，一个有helper节点的工作流 | |  |  | |  | |  | |  |
| 3 | 设置helper的消息回调函数 | |  |  | |  | |  | |  |
| 4 | 部署工作流 | |  |  | |  | |  | |  |
| 5 | 检查节点工作状态 | | 请求1 | feishu-msg节点返回预期消息1 | | feishu-msg节点返回预期消息1 | | 返回的消息与预期是否相等 | | 通过 |
| 6 | 停止node-red测试环境 | |  |  | |  | |  | |  |
| **备注** | | 请求1  {  "uuid": "41b5f371157e3d5341b38b20396e77e3",  "token": "2g7als3DgPW6Xp1xEpmcvgVhQG621bFY",//校验Token  "ts": "1550038209.428520", //时间戳  "type": "event\_callback",//事件回调此处固定为event\_callback  "event": {  "type": "message", // 事件类型  "app\_id": "cli\_xxx",  "tenant\_key": "xxx", //企业标识  "root\_id": "",  "parent\_id": "",  "open\_chat\_id": TEST\_open\_chat\_id,  "chat\_type": "group",//私聊private，群聊group  "msg\_type": "image", //消息类型  "open\_id": TEST\_open\_id,  "open\_message\_id": "om\_36686ee62209da697d8775375d0c8e88",  "is\_mention": false,  "image\_key": TEST\_IMAGE\_KEY  }  })  预期消息1  {  payload: {  msg: {  image\_key: TEST\_IMAGE\_KEY  },  },  feishu\_meta\_info: {  target: {  chat: [TEST\_open\_chat\_id],  user: []  },  chat: [TEST\_open\_chat\_id],  user: []  }  } | | | | | | | | |

## 创建飞书获得上传图片节点测试

* + 1. 飞书获得上传图片节点加载测试

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **测试用例名称** | | feishu-upload-image  节点加载测试 | **测试用例标识** | | TC010 | | **测试需求标识** | | TR2.6 | |
| **简要描述** | | 测试 feishu-upload-image节点是否能正确加载 | | | | | | | | |
| **前提和约束** | | 无 | | | | | | | | |
| **测试方法** | | 冒烟测试 | | | | | | | | |
| **测试过程描述** | | | | | | | | | | |
| **序号** | **测试步骤** | | **输入** | **测试结果** | | | | **评价准则** | | **测试结论** |
| **预期结果** | | **实际结果** | |
| 1 | 搭建node-red测试环境 | |  |  | |  | |  | |  |
| 2 | 创建含有一个feishu-upload-image节点和一个feishu-config节点的工作流 | |  |  | |  | |  | |  |
| 3 | 按照飞书tenant\_access\_token API 架设nock拦截器 | |  |  | |  | |  | |  |
| 4 | 部署工作流 | |  |  | |  | |  | |  |
| 5 | 检查节点加载状态 | |  | feishu-config节点获得tenant\_token | | feishu-config节点获得tenant\_token | | 检查tenant\_token是否与预设值相同 | | 通过 |
|  | feishu-upload-image节点没有报错 | | feishu-upload-image节点没有报错 | | 检查是否有未处理的异常 | | 通过 |
| 6 | 停止node-red测试环境 | |  |  | |  | |  | |  |
| **备注** | | 无 | | | | | | | | |

* + 1. 飞书获得上传图片节点功能测试

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **测试用例名称** | | feishu-upload-image  节点上传图片测试 | **测试用例标识** | | TC011 | | **测试需求标识** | | TR2.6 | |
| **简要描述** | | 测试 feishu-upload-image节点是否能正确上传图片 | | | | | | | | |
| **前提和约束** | | 无 | | | | | | | | |
| **测试方法** | | 单元测试 | | | | | | | | |
| **测试过程描述** | | | | | | | | | | |
| **序号** | **测试步骤** | | **输入** | **测试结果** | | | | **评价准则** | | **测试结论** |
| **预期结果** | | **实际结果** | |
| 1 | 搭建node-red测试环境 | |  |  | |  | |  | |  |
| 2 | 创建含有一个feishu-upload-image节点和一个feishu-config节点的工作流 | |  |  | |  | |  | |  |
| 3 | 按照飞书tenant\_access\_token API 架设nock拦截器 | |  |  | |  | |  | |  |
| 4 | 按照飞书upload\_image API架设nock拦截器 | |  |  | |  | |  | |  |
| 5 | Helper节点上设置回调监听，监听upload的返回 | |  |  | |  | |  | |  |
| 6 | 部署工作流 | |  |  | |  | |  | |  |
| 7 | 检查节点输入输出 | |  | feishu-config节点获得tenant\_token | | feishu-config节点获得tenant\_token | | 检查tenant\_token是否与预设值相同 | | 通过 |
|  | 拦截器收到合法请求 | | 拦截器收到合法请求 | | 上传的Buffer是否与预设的相同 | | 通过 |
|  | Helper收到预设的TEST\_IMAGE\_KEY | | Helper收到预设的TEST\_IMAGE\_KEY | | 返回的image\_key与预设是否一致 | | 通过 |
| 8 | 停止node-red测试环境 | |  |  | |  | |  | |  |
| **备注** | |  | | | | | | | | |

## 创建飞书获取图片节点测试

* + 1. 飞书获取图片节点加载测试

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **测试用例名称** | | feishu-fetch-image  节点加载测试 | **测试用例标识** | | TC012 | | **测试需求标识** | | TR2.7 | |
| **简要描述** | | 测试 feishu-fetch-image节点是否能正确加载 | | | | | | | | |
| **前提和约束** | | 无 | | | | | | | | |
| **测试方法** | | 冒烟测试 | | | | | | | | |
| **测试过程描述** | | | | | | | | | | |
| **序号** | **测试步骤** | | **输入** | **测试结果** | | | | **评价准则** | | **测试结论** |
| **预期结果** | | **实际结果** | |
| 1 | 搭建node-red测试环境 | |  |  | |  | |  | |  |
| 2 | 创建含有一个feishu-fetch-image节点和一个feishu-config节点的工作流 | |  |  | |  | |  | |  |
| 3 | 按照飞书tenant\_access\_token API 架设nock拦截器 | |  |  | |  | |  | |  |
| 4 | 部署工作流 | |  |  | |  | |  | |  |
| 5 | 检查节点加载状态 | |  | feishu-config节点获得tenant\_token | | feishu-config节点获得tenant\_token | | 检查tenant\_token是否与预设值相同 | | 通过 |
|  | feishu-fetch-image节点没有报错 | | feishu-fetch-image节点没有报错 | | 检查是否有未处理的异常 | | 通过 |
| 6 | 停止node-red测试环境 | |  |  | |  | |  | |  |
| **备注** | | 无 | | | | | | | | |

* + 1. 飞书获取图片节点功能测试

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **测试用例名称** | | feishu-fetch-image  节点功能测试 | **测试用例标识** | | TC013 | | **测试需求标识** | | TR2.7 | |
| **简要描述** | | 测试 feishu-fetch-image节点是否能正确获取图片 | | | | | | | | |
| **前提和约束** | | 无 | | | | | | | | |
| **测试方法** | | 单元测试 | | | | | | | | |
| **测试过程描述** | | | | | | | | | | |
| **序号** | **测试步骤** | | **输入** | **测试结果** | | | | **评价准则** | | **测试结论** |
| **预期结果** | | **实际结果** | |
| 1 | 搭建node-red测试环境 | |  |  | |  | |  | |  |
| 2 | 创建含有一个feishu-fetch-image节点一个feishu-config节点,和一个helper节点的工作流 | |  |  | |  | |  | |  |
| 3 | 按照飞书tenant\_access\_token API 架设nock拦截器 | |  |  | |  | |  | |  |
| 4 | 按照飞书获取图片API架设nock拦截器 | |  |  | |  | |  | |  |
| 5 | 设置helper节点的接收消息回调 | |  |  | |  | |  | |  |
| 6 | 部署工作流 | |  |  | |  | |  | |  |
| 7 | 检查节点输入输出状态 | |  | feishu-config节点获得tenant\_token | | feishu-config节点获得tenant\_token | | 检查tenant\_token是否与预设值相同 | | 通过 |
| 预设的image\_key | Helper节点接收到预设的Buffer | | Helper节点接收到预设的Buffer | | 检查预设的Buffer是和收到的Buffer是否一致 | | 通过 |
| 8 | 停止node-red测试环境 | |  |  | |  | |  | |  |
| **备注** | |  | | | | | | | | |

## 创建飞书添加目标节点测试

* + 1. 飞书添加目标节点部署测试

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **测试用例名称** | | 飞书添加目标节点部署测试 | **测试用例标识** | | TC014 | | **测试需求标识** | | TR2.8 | |
| **简要描述** | | 测试飞书添加目标节点能否正常部署 | | | | | | | | |
| **前提和约束** | | 无 | | | | | | | | |
| **测试方法** | | 冒烟测试 | | | | | | | | |
| **测试过程描述** | | | | | | | | | | |
| **序号** | **测试步骤** | | **输入** | **测试结果** | | | | **评价准则** | | **测试结论** |
| **预期结果** | | **实际结果** | |
| 1 | 搭建node-red测试环境 | |  |  | |  | |  | |  |
| 2 | 创建一个含有无配置的飞书添加目标节点的工作流，name属性为test | |  |  | |  | |  | |  |
| 3 | 部署工作流 | |  |  | |  | |  | |  |
| 4 | 获取飞书目标节点的name属性 | |  | test | | test | | 内容一致 | | 通过 |
| 5 | 停止node-red测试环境 | |  |  | |  | |  | |  |
| **备注** | | 无 | | | | | | | | |

* + 1. 飞书添加目标节点正则表达式筛选器测试

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **测试用例名称** | | 飞书添加目标节点正则表达式筛选器测试 | **测试用例标识** | | TC015 | | **测试需求标识** | | TR2.8 | |
| **简要描述** | | 测试飞书添加目标节点是否能依据给定的正则表达式添加特定的发送目标 | | | | | | | | |
| **前提和约束** | | 无 | | | | | | | | |
| **测试方法** | | 单元测试 | | | | | | | | |
| **测试过程描述** | | | | | | | | | | |
| **序号** | **测试步骤** | | **输入** | **测试结果** | | | | **评价准则** | | **测试结论** |
| **预期结果** | | **实际结果** | |
| 1 | 搭建node-red测试环境 | |  |  | |  | |  | |  |
| 2 | 创建含有一个飞书添加目标节点节点和一个helper节点的工作流。 | | 飞书添加目标节点配置见备注。 |  | |  | |  | |  |
| 3 | 设置helper节点的消息处理回调函数用于后续判断 | |  |  | |  | |  | |  |
| 4 | 修改飞书添加目标节点节点正则表达式及其选项，部署工作流后，输入数据并比对结果 | | 需修改配置：   * 正则表达式：“id\_1” * 是否忽略大小写：否   输入数据见备注 | helper节点收到数据的 feishu\_meta\_info.target.user 域的长度为 0；feishu\_meta\_info.target.chat的长度为1；feishu\_meta\_info.target.chat[1]为"test\_chat\_id\_1" | | helper节点收到数据的 feishu\_meta\_info.target.user 域的长度为 0；feishu\_meta\_info.target.chat的长度为1；feishu\_meta\_info.target.chat[1]为"test\_chat\_id\_1" | | 同预期结果一致 | | 通过 |
| 需修改配置：   * 正则表达式：“Id\_2” * 是否忽略大小写：是   输入数据见备注 | helper节点收到数据的 feishu\_meta\_info.target.user 域的长度为 0；feishu\_meta\_info.target.chat的长度为1；feishu\_meta\_info.target.chat[0]为"test\_chat\_id\_2" | | helper节点收到数据的 feishu\_meta\_info.target.user 域的长度为 0；feishu\_meta\_info.target.chat的长度为1；feishu\_meta\_info.target.chat[0]为"test\_chat\_id\_2" | | 同预期结果一致 | | 通过 |
| 5 | 停止node-red测试环境 | |  |  | |  | |  | |  |
| **备注** | | 节点配置：  {  addMode: 'filter',  directAddProperty: 'group',  directAddValueList: [],  evalPropertyClass: 'group',  evalGroupProperty: 'open\_chat\_id',  evalUserProperty: 'open\_user\_id',  filterMode: 'regex',  filterValueList: [ '' ],  filterRegexPattern: 'id\_1',  filterRegexIgnoreCase: false  }  输入数据：  {  payload: 'test\_message',  feishu\_meta\_info: { chat: [ 'test\_chat\_id\_1', 'TEST\_CHAT\_ID\_1', 'test\_chat\_id\_2' ], user: [] }  } | | | | | | | | |

* + 1. 飞书添加目标节点列表筛选器测试

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **测试用例名称** | | 飞书添加目标节点列表筛选器测试 | **测试用例标识** | | TC016 | | **测试需求标识** | | TR2.8 | |
| **简要描述** | | 测试飞书添加目标节点是否能依据给定的列表添加特定的发送目标 | | | | | | | | |
| **前提和约束** | | 无 | | | | | | | | |
| **测试方法** | | 单元测试 | | | | | | | | |
| **测试过程描述** | | | | | | | | | | |
| **序号** | **测试步骤** | | **输入** | **测试结果** | | | | **评价准则** | | **测试结论** |
| **预期结果** | | **实际结果** | |
| 1 | 搭建node-red测试环境 | |  |  | |  | |  | |  |
| 2 | 创建含有一个飞书添加目标节点节点和一个helper节点的工作流。 | | 飞书添加目标节点配置见备注。 |  | |  | |  | |  |
| 3 | 设置helper节点的消息处理回调函数用于后续判断 | |  |  | |  | |  | |  |
| 4 | 部署工作流 | |  |  | |  | |  | |  |
| 5 | 输入数据并比对结果 | | 输入数据见备注 | helper节点收到数据的 feishu\_meta\_info.target.user 域的长度为 0；feishu\_meta\_info.target.chat的长度为1；feishu\_meta\_info.target.chat[0]为"test\_chat\_id\_1" | | helper节点收到数据的 feishu\_meta\_info.target.user 域的长度为 0；feishu\_meta\_info.target.chat的长度为1；feishu\_meta\_info.target.chat[0]为"test\_chat\_id\_1" | | 同预期结果一致 | | 通过 |
| 6 | 停止node-red测试环境 | |  |  | |  | |  | |  |
| **备注** | | 节点配置：  {  addMode: 'filter',  directAddProperty: 'group',  directAddValueList: [],  evalPropertyClass: 'group',  evalGroupProperty: 'open\_chat\_id',  evalUserProperty: 'open\_user\_id',  filterMode: 'set',  filterValueList: [ 'test\_chat\_id\_1' ],  filterRegexPattern: '',  filterRegexIgnoreCase: false  }  输入数据：  {  payload: 'test\_message',  feishu\_meta\_info: { chat: [ 'test\_chat\_id\_1', 'test\_chat\_id\_2' ], user: [] }  } | | | | | | | | |

* + 1. 飞书添加目标节点筛选对象及属性测试

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **测试用例名称** | | 飞书添加目标节点筛选对象及属性测试 | **测试用例标识** | | TC017 | | **测试需求标识** | | TR2.8 | |
| **简要描述** | | 测试飞书添加目标节点是否能依据给定的筛选对象及筛选属性添加特定的发送目标 | | | | | | | | |
| **前提和约束** | | 无 | | | | | | | | |
| **测试方法** | | 单元测试 | | | | | | | | |
| **测试过程描述** | | | | | | | | | | |
| **序号** | **测试步骤** | | **输入** | **测试结果** | | | | **评价准则** | | **测试结论** |
| **预期结果** | | **实际结果** | |
| 1 | 搭建node-red测试环境 | |  |  | |  | |  | |  |
| 2 | 创建含有一个飞书添加目标节点节点和一个helper节点的工作流。 | | 飞书添加目标节点配置见备注。 |  | |  | |  | |  |
| 3 | 设置helper节点的消息处理回调函数用于后续判断 | |  |  | |  | |  | |  |
| 4 | 修改飞书添加目标节点节点筛选对象，部署工作流后，输入数据并比对结果 | | 需修改配置：   * 筛选对象：“group” * 筛选属性：“open\_chat\_id”   输入数据1,见备注 | helper节点收到数据的 feishu\_meta\_info.target.user 域的长度为 0；feishu\_meta\_info.target.chat的长度为1；feishu\_meta\_info.target.chat[0]为"test\_value" | | helper节点收到数据的 feishu\_meta\_info.target.user 域的长度为 0；feishu\_meta\_info.target.chat的长度为1；feishu\_meta\_info.target.chat[0]为"test\_value" | | 同预期结果一致 | | 通过 |
| 需修改配置：   * 筛选对象：“group” * 筛选属性：“open\_chat\_id”   输入数据2,见备注 | helper节点收到数据的 feishu\_meta\_info.target.user 域的长度为 0；feishu\_meta\_info.target.chat的长度为1；feishu\_meta\_info.target.chat[0]为{chat\_id: " test\_value"} | | helper节点收到数据的 feishu\_meta\_info.target.user 域的长度为 0；feishu\_meta\_info.target.chat的长度为1；feishu\_meta\_info.target.chat[0]为{chat\_id: " test\_value"} | | 同预期结果一致 | | 通过 |
| 需修改配置：   * 筛选对象：“group” * 筛选属性：“name”   输入数据3,见备注 | helper节点收到数据的 feishu\_meta\_info.target.user 域的长度为 0；feishu\_meta\_info.target.chat的长度为0 | | helper节点收到数据的 feishu\_meta\_info.target.user 域的长度为 0；feishu\_meta\_info.target.chat的长度为0 | | 同预期结果一致 | | 通过 |
| 需修改配置：   * 筛选对象：“group” * 筛选属性：“name”   输入数据4,见备注 | helper节点收到数据的 feishu\_meta\_info.target.user 域的长度为 0；feishu\_meta\_info.target.chat的长度为1；feishu\_meta\_info.target.chat[0]为{chat\_id: "1", name: "test\_value"} | | helper节点收到数据的 feishu\_meta\_info.target.user 域的长度为 0；feishu\_meta\_info.target.chat的长度为1；feishu\_meta\_info.target.chat[0]为{chat\_id: "1", name: "test\_value"} | | 同预期结果一致 | | 通过 |
| 需修改配置：   * 筛选对象：“user” * 筛选属性：“user\_open\_id”   输入数据5,见备注 | helper节点收到数据的 feishu\_meta\_info.target.user 域的长度为 1；feishu\_meta\_info.target.chat的长度为0；feishu\_meta\_info.target.chat[0]为"test\_value" | | helper节点收到数据的 feishu\_meta\_info.target.user 域的长度为 1；feishu\_meta\_info.target.chat的长度为0；feishu\_meta\_info.target.chat[0]为"test\_value" | | 同预期结果一致 | | 通过 |
| 5 | 停止node-red测试环境 | |  |  | |  | |  | |  |
| **备注** | | 节点配置：  {  addMode: 'filter',  directAddProperty: 'group',  directAddValueList: [],  evalPropertyClass: 'group',  evalGroupProperty: 'open\_chat\_id',  evalUserProperty: 'open\_user\_id',  filterMode: 'set',  filterValueList: [ 'test\_value' ],  filterRegexPattern: '',  filterRegexIgnoreCase: false  }  输入数据1：  {  payload: 'test\_message',  feishu\_meta\_info: { chat: [ 'test\_value', 'not\_test\_value' ], user: [] }  }  输入数据2：  {  payload: 'test\_message',  feishu\_meta\_info: {  chat: [ { chat\_id: 'test\_value' }, { chat\_id: 'not\_test\_value' } ],  user: []  }  }  输入数据3：  {  payload: 'test\_message',  feishu\_meta\_info: { chat: [ 'test\_value' ], user: [] }  }  输入数据4：  {  payload: 'test\_message',  feishu\_meta\_info: {  chat: [  { chat\_id: '1', name: 'test\_value' },  { chat\_id: '2', name: 'not\_test\_value' }  ],  user: []  }  }  输入数据5:  {  payload: 'test\_message',  feishu\_meta\_info: { chat: [], user: [ 'test\_value', 'not\_test\_value' ] }  } | | | | | | | | |

* + 1. 飞书添加目标节点直接添加模式测试

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **测试用例名称** | | 飞书添加目标节点直接添加模式测试 | **测试用例标识** | | TC018 | | **测试需求标识** | | TR2.8 | |
| **简要描述** | | 测试飞书添加目标节点是否能添加特定的发送目标 | | | | | | | | |
| **前提和约束** | | 无 | | | | | | | | |
| **测试方法** | | 单元测试 | | | | | | | | |
| **测试过程描述** | | | | | | | | | | |
| **序号** | **测试步骤** | | **输入** | **测试结果** | | | | **评价准则** | | **测试结论** |
| **预期结果** | | **实际结果** | |
| 1 | 搭建node-red测试环境 | |  |  | |  | |  | |  |
| 2 | 创建含有一个飞书添加目标节点节点和一个helper节点的工作流。 | | 飞书添加目标节点配置见备注。 |  | |  | |  | |  |
| 3 | 设置helper节点的消息处理回调函数用于后续判断 | |  |  | |  | |  | |  |
| 4 | 修改飞书添加目标节点节点正则表达式及其选项，部署工作流后，输入数据并比对结果 | | 需修改配置：   * 添加对象：“group”   输入数据见备注 | helper节点收到数据的 feishu\_meta\_info.target.user 域的长度为 0；feishu\_meta\_info.target.chat的长度为1；feishu\_meta\_info.target.chat[1]为"test\_value" | | helper节点收到数据的 feishu\_meta\_info.target.user 域的长度为 0；feishu\_meta\_info.target.chat的长度为1；feishu\_meta\_info.target.chat[1]为"test\_value" | | 同预期结果一致 | | 通过 |
| 需修改配置：   * 添加对象：“user”   输入数据见备注 | helper节点收到数据的 feishu\_meta\_info.target.user 域的长度为 1；feishu\_meta\_info.target.chat的长度为0；feishu\_meta\_info.target.user[0]为"test\_value" | | helper节点收到数据的 feishu\_meta\_info.target.user 域的长度为 1；feishu\_meta\_info.target.chat的长度为0；feishu\_meta\_info.target.user[0]为"test\_value" | | 同预期结果一致 | | 通过 |
| 5 | 停止node-red测试环境 | |  |  | |  | |  | |  |
| **备注** | | 节点配置：  {  addMode: 'direct',  directAddValueList: [ 'test\_value' ],  evalPropertyClass: 'group',  evalGroupProperty: 'open\_chat\_id',  evalUserProperty: 'open\_user\_id',  filterMode: 'regex',  filterValueList: [ '' ],  filterRegexPattern: '',  filterRegexIgnoreCase: false,  directAddProperty: 'group'  }  输入数据：  { payload: 'test\_message', feishu\_meta\_info: {} } | | | | | | | | |

* + 1. 飞书添加目标节点初始化测试

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **测试用例名称** | | 飞书添加目标节点初始化测试 | **测试用例标识** | | TC019 | | **测试需求标识** | | TR2.8 | |
| **简要描述** | | 测试飞书添加目标节点是否能在输入不存在所需域时对相应域进行初始化 | | | | | | | | |
| **前提和约束** | | 无 | | | | | | | | |
| **测试方法** | | 单元测试 | | | | | | | | |
| **测试过程描述** | | | | | | | | | | |
| **序号** | **测试步骤** | | **输入** | **测试结果** | | | | **评价准则** | | **测试结论** |
| **预期结果** | | **实际结果** | |
| 1 | 搭建node-red测试环境 | |  |  | |  | |  | |  |
| 2 | 创建含有一个飞书添加目标节点节点和一个helper节点的工作流。 | | 飞书添加目标节点配置见备注。 |  | |  | |  | |  |
| 3 | 设置helper节点的消息处理回调函数用于后续判断 | |  |  | |  | |  | |  |
| 4 | 部署工作流 | |  |  | |  | |  | |  |
| 5 | 输入数据并比对结果 | | 输入数据1，见备注 | helper节点收到数据的 feishu\_meta\_info 域为期望值，见备注 | | helper节点收到数据的 feishu\_meta\_info 域为期望值，见备注 | | 同预期结果一致 | | 通过 |
| 输入数据2，见备注 | helper节点收到数据的 feishu\_meta\_info 域为期望值，见备注 | | helper节点收到数据的 feishu\_meta\_info 域为期望值，见备注 | | 同预期结果一致 | | 通过 |
| 输入数据3，见备注 | helper节点收到数据的 feishu\_meta\_info 域为期望值，见备注 | | helper节点收到数据的 feishu\_meta\_info 域为期望值，见备注 | | 同预期结果一致 | | 通过 |
| 输入数据4，见备注 | helper节点收到数据的 feishu\_meta\_info 域为期望值，见备注 | | helper节点收到数据的 feishu\_meta\_info 域为期望值，见备注 | | 同预期结果一致 | | 通过 |
| 6 | 停止node-red测试环境 | |  |  | |  | |  | |  |
| **备注** | | 节点配置：  {  addMode: 'direct',  directAddProperty: 'group',  directAddValueList: [],  evalPropertyClass: 'group',  evalGroupProperty: 'open\_chat\_id',  evalUserProperty: 'open\_user\_id',  filterMode: 'regex',  filterValueList: [ '' ],  filterRegexPattern: '',  filterRegexIgnoreCase: false  }  输入数据1：  { payload: 'test\_message' }  输入数据2：  { payload: 'test\_message', feishu\_meta\_info: {} }  输入数据3：  { payload: 'test\_message', feishu\_meta\_info: { target: { user: [] } } }  输入数据4：  { payload: 'test\_message', feishu\_meta\_info: { target: { chat: [] } } }  feishu\_meta\_info 域期望值：  {  target: {  chat: [],  user: []  }  } | | | | | | | | |

* + 1. 飞书添加目标节点重复添加测试

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **测试用例名称** | | 飞书添加目标节点重复添加测试 | **测试用例标识** | | TC020 | | **测试需求标识** | | TR2.8 | |
| **简要描述** | | 测试飞书添加目标节点是否能避免将重复目标添加至发送目标中 | | | | | | | | |
| **前提和约束** | | 无 | | | | | | | | |
| **测试方法** | | 单元测试 | | | | | | | | |
| **测试过程描述** | | | | | | | | | | |
| **序号** | **测试步骤** | | **输入** | **测试结果** | | | | **评价准则** | | **测试结论** |
| **预期结果** | | **实际结果** | |
| 1 | 搭建node-red测试环境 | |  |  | |  | |  | |  |
| 2 | 创建含有一个飞书添加目标节点节点和一个helper节点的工作流。 | | 飞书添加目标节点配置见备注。 |  | |  | |  | |  |
| 3 | 设置helper节点的消息处理回调函数用于后续判断 | |  |  | |  | |  | |  |
| 4 | 修改节点配置，部署工作流，输入数据并比对结果 | | 需修改配置：   * 添加对象：“group”   输入数据1，见备注 | helper节点收到数据的 feishu\_meta\_info 域为期望值1，见备注 | | helper节点收到数据的 feishu\_meta\_info 域为期望值1，见备注 | | 同预期结果一致 | | 通过 |
| 需修改配置：   * 添加对象：“user”   输入数据2，见备注 | helper节点收到数据的 feishu\_meta\_info 域为期望值2，见备注 | | helper节点收到数据的 feishu\_meta\_info 域为期望值2，见备注 | | 同预期结果一致 | | 通过 |
| 5 | 停止node-red测试环境 | |  |  | |  | |  | |  |
| **备注** | | 节点配置：  {  addMode: 'direct',  directAddProperty: 'group',  directAddValueList: [ 'test\_value' ],  evalPropertyClass: 'group',  evalGroupProperty: 'open\_chat\_id',  evalUserProperty: 'open\_user\_id',  filterMode: 'regex',  filterValueList: [ '' ],  filterRegexPattern: '',  filterRegexIgnoreCase: false  }  输入数据1：  {  payload: 'test\_message',  feishu\_meta\_info: {  target: { chat: [ { chat\_id: 'test\_value' } ], user: [] }  }  }  输入数据2：  {  payload: 'test\_message',  feishu\_meta\_info: { target: { chat: [], user: [ 'test\_value' ] } }  }  feishu\_meta\_info 域期望值1：  {  target: {  chat: [{chat\_id: "test\_value"}],  user: []  }  }  feishu\_meta\_info 域期望值2：  {  target: {  chat: [],  user: ["test\_value"]  }  } | | | | | | | | |

* + 1. 飞书添加目标节点异常处理测试

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **测试用例名称** | | 飞书添加目标节点异常处理测试 | **测试用例标识** | | TC021 | | **测试需求标识** | | TR2.8 | |
| **简要描述** | | 测试飞书添加目标节点是否能在输入不符合预期的情况下正常运行并报告相应错误 | | | | | | | | |
| **前提和约束** | | 无 | | | | | | | | |
| **测试方法** | | 单元测试 | | | | | | | | |
| **测试过程描述** | | | | | | | | | | |
| **序号** | **测试步骤** | | **输入** | **测试结果** | | | | **评价准则** | | **测试结论** |
| **预期结果** | | **实际结果** | |
| 1 | 搭建node-red测试环境 | |  |  | |  | |  | |  |
| 2 | 创建含有一个飞书添加目标节点节点和一个helper节点的工作流。 | | 飞书添加目标节点配置见备注。 |  | |  | |  | |  |
| 3 | 设置helper节点的消息处理回调函数用于后续判断 | |  |  | |  | |  | |  |
| 4 | 修改节点配置，部署工作流，输入数据并比对结果 | | 需修改配置：   * 添加模式：“filter” * 筛选对象：“group” * 筛选属性：“open\_chat\_id”   输入数据1，见备注 | feishu\_meta\_info.target 为 { chat: [], user: [] }；节点报告警告“Expect msg to have feishu\_meta\_info.chat field.” | | feishu\_meta\_info.target 为 { chat: [], user: [] }；节点报告警告“Expect msg to have feishu\_meta\_info.chat field.” | | 同预期结果一致 | | 通过 |
| 需修改配置：   * 添加模式：“filter” * 筛选对象：“user” * 筛选属性：“open\_user\_id”   输入数据1，见备注 | feishu\_meta\_info.target 为 { chat: [], user: [] }；节点报告警告“Expect msg to have feishu\_meta\_info.user field.” | | feishu\_meta\_info.target 为 { chat: [], user: [] }；节点报告警告“Expect msg to have feishu\_meta\_info.user field.” | | 同预期结果一致 | | 通过 |
| 需修改配置：   * 添加模式：“filter” * 筛选对象：“group” * 筛选属性：“open\_chat\_id”   输入数据2，见备注 | feishu\_meta\_info.target.chat的长度为1；feishu\_meta\_info.target.chat[0]为“test\_value”；节点报告错误“Wrong chat info structure detected” | | feishu\_meta\_info.target.chat的长度为1；feishu\_meta\_info.target.chat[0]为“test\_value”；节点报告错误“Wrong chat info structure detected” | | 同预期结果一致 | | 通过 |
| 需修改配置：   * 添加模式：“filter” * 筛选对象：“group” * 筛选属性：“name”   输入数据3，见备注 | feishu\_meta\_info.target.chat的长度为1；feishu\_meta\_info.target.chat[0]为{ chat\_id: "test\_chat\_id\_1", name: "test\_value" }；节点报告错误“Wrong chat info structure detected” | | feishu\_meta\_info.target.chat的长度为1；feishu\_meta\_info.target.chat[0]为{ chat\_id: "test\_chat\_id\_1", name: "test\_value" }；节点报告错误“Wrong chat info structure detected” | | 同预期结果一致 | | 通过 |
| 需修改配置：   * 添加模式：“filter” * 筛选对象：“group” * 筛选属性：“name”   输入数据4，见备注 | feishu\_meta\_info.target.chat的长度为0；节点报告错误“Wrong chat info structure detected” | | feishu\_meta\_info.target.chat的长度为0；节点报告错误“Wrong chat info structure detected” | | 同预期结果一致 | | 通过 |
| 需修改配置：   * 添加模式：“filter” * 筛选对象：“user” * 筛选属性：“open\_user\_id”   输入数据5，见备注 | feishu\_meta\_info.target.user的长度为1；feishu\_meta\_info.target.user[0]为“test\_value”；节点报告错误“Wrong user info structure detected” | | feishu\_meta\_info.target.user的长度为1；feishu\_meta\_info.target.user[0]为“test\_value”；节点报告错误“Wrong user info structure detected” | | 同预期结果一致 | | 通过 |
| 需修改配置：   * 添加模式：“direct” * 添加对象：“group”   输入数据6，见备注 | feishu\_meta\_info.target.chat的长度为2；feishu\_meta\_info.target.chat[0]为{name: "test\_value"}；feishu\_meta\_info.target.chat[1]为“test\_value”；节点报告错误“Wrong chat info structure detected” | | feishu\_meta\_info.target.chat的长度为2；feishu\_meta\_info.target.chat[0]为{name: "test\_value"}；feishu\_meta\_info.target.chat[1]为“test\_value”；节点报告错误“Wrong chat info structure detected” | | 同预期结果一致 | | 通过 |
| 需修改配置：   * 添加模式：“direct” * 添加对象：“user”   输入数据7，见备注 | feishu\_meta\_info.target.user的长度为2；feishu\_meta\_info.target.user[0]为{}；feishu\_meta\_info.target.user[1]为“test\_value”；节点报告错误“Wrong user info structure detected” | | feishu\_meta\_info.target.user的长度为2；feishu\_meta\_info.target.user[0]为{}；feishu\_meta\_info.target.user[1]为“test\_value”；节点报告错误“Wrong user info structure detected” | | 同预期结果一致 | | 通过 |
| 5 | 停止node-red测试环境 | |  |  | |  | |  | |  |
| **备注** | | 节点配置：  {  addMode: 'filter',  directAddValueList: [ 'test\_value' ],  evalPropertyClass: 'group',  evalGroupProperty: 'open\_chat\_id',  evalUserProperty: 'open\_user\_id',  filterMode: 'regex',  filterValueList: [ '' ],  filterRegexPattern: 'test\_value',  filterRegexIgnoreCase: false  }  输入数据1：  { payload: 'test\_message', feishu\_meta\_info: {} }  输入数据2：  {  payload: 'test\_message',  feishu\_meta\_info: { chat: [ {}, 'test\_value' ], user: [] }  }  输入数据3：  {  payload: 'test\_message',  feishu\_meta\_info: {  chat: [ {}, { chat\_id: 'test\_chat\_id\_1', name: 'test\_value' } ],  user: []  }  }  输入数据4：  {  payload: 'test\_message',  feishu\_meta\_info: { chat: [ { name: 'test\_value' } ], user: [] }  }  输入数据5：  {  payload: 'test\_message',  feishu\_meta\_info: { chat: [], user: [ {}, 'test\_value' ] }  }  输入数据6：  {  payload: 'test\_message',  feishu\_meta\_info: {  target: { chat: [ { name: 'test\_value' } ], user: [] }  }  }  输入数据7：  {  payload: 'test\_message',  feishu\_meta\_info: { target: { chat: [], user: [ {} ] } }  } | | | | | | | | |

* + 1. 飞书添加目标节点配置异常处理测试

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **测试用例名称** | | 飞书添加目标节点配置异常处理测试 | **测试用例标识** | | TC022 | | **测试需求标识** | | TR2.8 | |
| **简要描述** | | 测试飞书添加目标节点是否能在配置不符合预期的情况下正常运行并报告相应错误 | | | | | | | | |
| **前提和约束** | | 无 | | | | | | | | |
| **测试方法** | | 单元测试 | | | | | | | | |
| **测试过程描述** | | | | | | | | | | |
| **序号** | **测试步骤** | | **输入** | **测试结果** | | | | **评价准则** | | **测试结论** |
| **预期结果** | | **实际结果** | |
| 1 | 搭建node-red测试环境 | |  |  | |  | |  | |  |
| 2 | 创建含有一个飞书添加目标节点节点和一个helper节点的工作流。 | | 飞书添加目标节点配置见备注。 |  | |  | |  | |  |
| 3 | 设置helper节点的消息处理回调函数用于后续判断 | |  |  | |  | |  | |  |
| 4 | 修改节点配置，部署工作流，输入数据并比对结果 | | 需修改配置：   * 添加模式：“invalid”   输入数据，见备注 | 节点报告错误“Unrecognized addMode: invalid”；  helper节点收到数据的 feishu\_meta\_info 域为期望值，见备注 | | 节点报告错误“Unrecognized addMode: invalid”；  helper节点收到数据的 feishu\_meta\_info 域为期望值，见备注 | | 同预期结果一致 | | 通过 |
| 需修改配置：   * 添加模式：“direct” * 添加对象：“invalid”   输入数据，见备注 | 节点报告错误“Unrecognized directAddProperty: invalid”；  helper节点收到数据的 feishu\_meta\_info 域为期望值，见备注 | | 节点报告错误“Unrecognized directAddProperty: invalid”；  helper节点收到数据的 feishu\_meta\_info 域为期望值，见备注 | | 同预期结果一致 | | 通过 |
| 需修改配置：   * 添加模式：“filter” * 筛选模式：“invalid”   输入数据，见备注 | 节点报告错误“Unrecognized filterMode: invalid”；  helper节点收到数据的 feishu\_meta\_info 域为期望值，见备注 | | 节点报告错误“Unrecognized filterMode: invalid”；  helper节点收到数据的 feishu\_meta\_info 域为期望值，见备注 | | 同预期结果一致 | | 通过 |
| 需修改配置：   * 添加模式：“filter” * 筛选对象：“invalid”   输入数据，见备注 | 节点报告错误“Unrecognized evalPropertyClass: invalid”；  helper节点收到数据的 feishu\_meta\_info 域为期望值，见备注 | | 节点报告错误“Unrecognized evalPropertyClass: invalid”；  helper节点收到数据的 feishu\_meta\_info 域为期望值，见备注 | | 同预期结果一致 | | 通过 |
| 需修改配置：   * 添加模式：“filter” * 筛选对象：“group” * 筛选属性：“invalid”   输入数据，见备注 | 节点报告错误“Unrecognized evalGroupProperty: invalid”；  helper节点收到数据的 feishu\_meta\_info 域为期望值，见备注 | | 节点报告错误“Unrecognized evalGroupProperty: invalid”；  helper节点收到数据的 feishu\_meta\_info 域为期望值，见备注 | | 同预期结果一致 | | 通过 |
| 需修改配置：   * 添加模式：“filter” * 筛选对象：“user” * 筛选属性：“invalid”   输入数据，见备注 | 节点报告错误“Unrecognized evalUserProperty: invalid”；  helper节点收到数据的 feishu\_meta\_info 域为期望值，见备注 | | 节点报告错误“Unrecognized evalUserProperty: invalid”；  helper节点收到数据的 feishu\_meta\_info 域为期望值，见备注 | | 同预期结果一致 | | 通过 |
| 5 | 停止node-red测试环境 | |  |  | |  | |  | |  |
| **备注** | | 节点配置：  {  addMode: "direct",  directAddProperty: "group",  directAddValueList: ["test\_value"],  evalPropertyClass: "group",  evalGroupProperty: "open\_chat\_id",  evalUserProperty: "open\_user\_id",  filterMode: "regex",  filterValueList: [""],  filterRegexPattern: "",  filterRegexIgnoreCase: false  }  输入数据：  {  payload: 'test\_message',  feishu\_meta\_info: { chat: [ 'test\_chat\_id' ], user: [ 'test\_user\_id' ] }  }  feishu\_meta\_info 域期望值：  {  target: {  chat: [],  user: []  },  chat: ["test\_chat\_id"],  user: ["test\_user\_id"]  } | | | | | | | | |

## 创建飞书移除目标节点测试

* + 1. 飞书移除目标节点部署测试

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **测试用例名称** | | 飞书移除目标节点部署测试 | **测试用例标识** | | TC023 | | **测试需求标识** | | TR2.9 | |
| **简要描述** | | 测试飞书移除目标节点能否正常部署 | | | | | | | | |
| **前提和约束** | | 无 | | | | | | | | |
| **测试方法** | | 冒烟测试 | | | | | | | | |
| **测试过程描述** | | | | | | | | | | |
| **序号** | **测试步骤** | | **输入** | **测试结果** | | | | **评价准则** | | **测试结论** |
| **预期结果** | | **实际结果** | |
| 1 | 搭建node-red测试环境 | |  |  | |  | |  | |  |
| 2 | 创建一个含有无配置的飞书移除目标节点的工作流，name属性为test | |  |  | |  | |  | |  |
| 3 | 部署工作流 | |  |  | |  | |  | |  |
| 4 | 获取飞书目标节点的name属性 | |  | test | | test | | 内容一致 | | 通过 |
| 5 | 停止node-red测试环境 | |  |  | |  | |  | |  |
| **备注** | | 无 | | | | | | | | |

* + 1. 飞书移除目标节点正则表达式筛选器测试

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **测试用例名称** | | 飞书移除目标节点正则表达式筛选器测试 | **测试用例标识** | | TC024 | | **测试需求标识** | | TR2.9 | |
| **简要描述** | | 测试飞书移除目标节点是否能依据给定的正则表达式对发送目标进行筛选 | | | | | | | | |
| **前提和约束** | | 无 | | | | | | | | |
| **测试方法** | | 单元测试 | | | | | | | | |
| **测试过程描述** | | | | | | | | | | |
| **序号** | **测试步骤** | | **输入** | **测试结果** | | | | **评价准则** | | **测试结论** |
| **预期结果** | | **实际结果** | |
| 1 | 搭建node-red测试环境 | |  |  | |  | |  | |  |
| 2 | 创建含有一个飞书移除目标节点节点和一个helper节点的工作流。 | | 飞书移除目标节点配置见备注。 |  | |  | |  | |  |
| 3 | 设置helper节点的消息处理回调函数用于后续判断 | |  |  | |  | |  | |  |
| 4 | 修改飞书移除目标节点节点正则表达式及其选项，部署工作流后，输入数据并比对结果 | | 需修改配置：   * 正则表达式：“id\_1” * 是否忽略大小写：否   输入数据见备注 | helper节点收到数据的 feishu\_meta\_info.target.user 域的长度为 0；feishu\_meta\_info.target.chat的长度为2；feishu\_meta\_info.target.chat[0]为"TEST\_CHAT\_ID\_1"；feishu\_meta\_info.target.chat[1]为"test\_chat\_id\_2" | | helper节点收到数据的 feishu\_meta\_info.target.user 域的长度为 0；feishu\_meta\_info.target.chat的长度为2；feishu\_meta\_info.target.chat[0]为"TEST\_CHAT\_ID\_1"；feishu\_meta\_info.target.chat[1]为"test\_chat\_id\_2" | | 同预期结果一致 | | 通过 |
| 需修改配置：   * 正则表达式：“id\_1” * 是否忽略大小写：是   输入数据见备注 | helper节点收到数据的 feishu\_meta\_info.target.user 域的长度为 0；feishu\_meta\_info.target.chat的长度为1；feishu\_meta\_info.target.chat[0]为"test\_chat\_id\_2" | | helper节点收到数据的 feishu\_meta\_info.target.user 域的长度为 0；feishu\_meta\_info.target.chat的长度为1；feishu\_meta\_info.target.chat[0]为"test\_chat\_id\_2" | | 同预期结果一致 | | 通过 |
| 5 | 停止node-red测试环境 | |  |  | |  | |  | |  |
| **备注** | | 节点配置：  {  evalPropertyClass: 'group',  evalGroupProperty: 'open\_chat\_id',  evalUserProperty: 'open\_user\_id',  filterMode: 'regex',  filterValueList: [ '' ],  filterRegexPattern: 'id\_1',  filterRegexIgnoreCase: false  }  输入数据：  {  payload: "test\_message",  feishu\_meta\_info: {  target: {  chat: [  "test\_chat\_id\_1",  "TEST\_CHAT\_ID\_1",  "test\_chat\_id\_2"  ],  user: []  }  }  } | | | | | | | | |

* + 1. 飞书移除目标节点列表筛选器测试

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **测试用例名称** | | 飞书移除目标节点列表筛选器测试 | **测试用例标识** | | TC025 | | **测试需求标识** | | TR2.9 | |
| **简要描述** | | 测试飞书移除目标节点是否能依据给定的列表对发送目标进行筛选 | | | | | | | | |
| **前提和约束** | | 无 | | | | | | | | |
| **测试方法** | | 单元测试 | | | | | | | | |
| **测试过程描述** | | | | | | | | | | |
| **序号** | **测试步骤** | | **输入** | **测试结果** | | | | **评价准则** | | **测试结论** |
| **预期结果** | | **实际结果** | |
| 1 | 搭建node-red测试环境 | |  |  | |  | |  | |  |
| 2 | 创建含有一个飞书移除目标节点节点和一个helper节点的工作流。 | | 飞书移除目标节点配置见备注。 |  | |  | |  | |  |
| 3 | 设置helper节点的消息处理回调函数用于后续判断 | |  |  | |  | |  | |  |
| 4 | 部署工作流 | |  |  | |  | |  | |  |
| 5 | 输入数据并比对结果 | | 输入数据见备注 | helper节点收到数据的 feishu\_meta\_info.target.user 域的长度为 0；feishu\_meta\_info.target.chat的长度为1；feishu\_meta\_info.target.chat[0]为"test\_chat\_id\_2" | | helper节点收到数据的 feishu\_meta\_info.target.user 域的长度为 0；feishu\_meta\_info.target.chat的长度为1；feishu\_meta\_info.target.chat[0]为"test\_chat\_id\_2" | | 同预期结果一致 | | 通过 |
| 6 | 停止node-red测试环境 | |  |  | |  | |  | |  |
| **备注** | | 节点配置：  {  evalPropertyClass: 'group',  evalGroupProperty: 'open\_chat\_id',  evalUserProperty: 'open\_user\_id',  filterMode: 'set',  filterValueList: [ 'test\_chat\_id\_1' ],  filterRegexPattern: '',  filterRegexIgnoreCase: false  }  输入数据：  {  payload: "test\_message",  feishu\_meta\_info: {  target: {  chat: [  "test\_chat\_id\_1",  "test\_chat\_id\_2"  ],  user: []  }  }  } | | | | | | | | |

* + 1. 飞书移除目标节点筛选对象及属性测试

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **测试用例名称** | | 飞书移除目标节点筛选对象及属性测试 | **测试用例标识** | | TC026 | | **测试需求标识** | | TR2.9 | |
| **简要描述** | | 测试飞书移除目标节点是否能依据给定的筛选对象及筛选属性对发送目标进行筛选 | | | | | | | | |
| **前提和约束** | | 无 | | | | | | | | |
| **测试方法** | | 单元测试 | | | | | | | | |
| **测试过程描述** | | | | | | | | | | |
| **序号** | **测试步骤** | | **输入** | **测试结果** | | | | **评价准则** | | **测试结论** |
| **预期结果** | | **实际结果** | |
| 1 | 搭建node-red测试环境 | |  |  | |  | |  | |  |
| 2 | 创建含有一个飞书移除目标节点节点和一个helper节点的工作流。 | | 飞书移除目标节点配置见备注。 |  | |  | |  | |  |
| 3 | 设置helper节点的消息处理回调函数用于后续判断 | |  |  | |  | |  | |  |
| 4 | 修改飞书移除目标节点节点筛选对象，部署工作流后，输入数据并比对结果 | | 需修改配置：   * 筛选对象：“group” * 筛选属性：“open\_chat\_id”   输入数据1,见备注 | helper节点收到数据的 feishu\_meta\_info.target.user 域的长度为 0；feishu\_meta\_info.target.chat的长度为1；feishu\_meta\_info.target.chat[0]为"not\_test\_value" | | helper节点收到数据的 feishu\_meta\_info.target.user 域的长度为 0；feishu\_meta\_info.target.chat的长度为1；feishu\_meta\_info.target.chat[0]为"not\_test\_value" | | 同预期结果一致 | | 通过 |
| 需修改配置：   * 筛选对象：“group” * 筛选属性：“open\_chat\_id”   输入数据2,见备注 | helper节点收到数据的 feishu\_meta\_info.target.user 域的长度为 0；feishu\_meta\_info.target.chat的长度为1；feishu\_meta\_info.target.chat[0]为{chat\_id: " not\_test\_value"} | | helper节点收到数据的 feishu\_meta\_info.target.user 域的长度为 0；feishu\_meta\_info.target.chat的长度为1；feishu\_meta\_info.target.chat[0]为{chat\_id: " not\_test\_value"} | | 同预期结果一致 | | 通过 |
| 需修改配置：   * 筛选对象：“group” * 筛选属性：“name”   输入数据3,见备注 | helper节点收到数据的 feishu\_meta\_info.target.user 域的长度为 0；feishu\_meta\_info.target.chat的长度为1；feishu\_meta\_info.target.chat[0]为"test\_value" | | helper节点收到数据的 feishu\_meta\_info.target.user 域的长度为 0；feishu\_meta\_info.target.chat的长度为1；feishu\_meta\_info.target.chat[0]为"test\_value" | | 同预期结果一致 | | 通过 |
| 需修改配置：   * 筛选对象：“group” * 筛选属性：“name”   输入数据4,见备注 | helper节点收到数据的 feishu\_meta\_info.target.user 域的长度为 0；feishu\_meta\_info.target.chat的长度为1；feishu\_meta\_info.target.chat[0]为{name: "not\_test\_value"} | | helper节点收到数据的 feishu\_meta\_info.target.user 域的长度为 0；feishu\_meta\_info.target.chat的长度为1；feishu\_meta\_info.target.chat[0]为{name: "not\_test\_value"} | | 同预期结果一致 | | 通过 |
| 需修改配置：   * 筛选对象：“user” * 筛选属性：“user\_open\_id”   输入数据5,见备注 | helper节点收到数据的 feishu\_meta\_info.target.user 域的长度为 1；feishu\_meta\_info.target.chat的长度为0；feishu\_meta\_info.target.chat[0]为"not\_test\_value" | | helper节点收到数据的 feishu\_meta\_info.target.user 域的长度为 1；feishu\_meta\_info.target.chat的长度为0；feishu\_meta\_info.target.chat[0]为"not\_test\_value" | | 同预期结果一致 | | 通过 |
| 5 | 停止node-red测试环境 | |  |  | |  | |  | |  |
| **备注** | | 节点配置：  {  evalPropertyClass: 'group',  evalGroupProperty: 'open\_chat\_id',  evalUserProperty: 'open\_user\_id',  filterMode: 'set',  filterValueList: [ 'test\_value' ],  filterRegexPattern: '',  filterRegexIgnoreCase: false  }  输入数据1：  {  payload: 'test\_message',  feishu\_meta\_info: { target: { chat: [ 'test\_value', 'not\_test\_value' ], user: [] } }  }  输入数据2：  {  payload: 'test\_message',  feishu\_meta\_info: {  target: {  chat: [ { chat\_id: 'test\_value' }, { chat\_id: 'not\_test\_value' } ],  user: []  }  }  }  输入数据3：  {  payload: 'test\_message',  feishu\_meta\_info: { target: { chat: [ 'test\_value' ], user: [] } }  }  输入数据4：  {  payload: 'test\_message',  feishu\_meta\_info: {  target: {  chat: [ { name: 'test\_value' }, { name: 'not\_test\_value' } ],  user: []  }  }  }  输入数据5:  {  payload: 'test\_message',  feishu\_meta\_info: { target: { chat: [], user: [ 'test\_value', 'not\_test\_value' ] } }  } | | | | | | | | |

* + 1. 飞书移除目标节点初始化测试

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **测试用例名称** | | 飞书移除目标节点初始化测试 | **测试用例标识** | | TC027 | | **测试需求标识** | | TR2.9 | |
| **简要描述** | | 测试飞书移除目标节点是否能在输入不存在所需域时对相应域进行初始化 | | | | | | | | |
| **前提和约束** | | 无 | | | | | | | | |
| **测试方法** | | 单元测试 | | | | | | | | |
| **测试过程描述** | | | | | | | | | | |
| **序号** | **测试步骤** | | **输入** | **测试结果** | | | | **评价准则** | | **测试结论** |
| **预期结果** | | **实际结果** | |
| 1 | 搭建node-red测试环境 | |  |  | |  | |  | |  |
| 2 | 创建含有一个飞书移除目标节点节点和一个helper节点的工作流。 | | 飞书移除目标节点配置见备注。 |  | |  | |  | |  |
| 3 | 设置helper节点的消息处理回调函数用于后续判断 | |  |  | |  | |  | |  |
| 4 | 部署工作流 | |  |  | |  | |  | |  |
| 5 | 输入数据并比对结果 | | 输入数据1，见备注 | helper节点收到数据的 feishu\_meta\_info 域为期望值，见备注 | | helper节点收到数据的 feishu\_meta\_info 域为期望值，见备注 | | 同预期结果一致 | | 通过 |
| 输入数据2，见备注 | helper节点收到数据的 feishu\_meta\_info 域为期望值，见备注 | | helper节点收到数据的 feishu\_meta\_info 域为期望值，见备注 | | 同预期结果一致 | | 通过 |
| 输入数据3，见备注 | helper节点收到数据的 feishu\_meta\_info 域为期望值，见备注 | | helper节点收到数据的 feishu\_meta\_info 域为期望值，见备注 | | 同预期结果一致 | | 通过 |
| 输入数据4，见备注 | helper节点收到数据的 feishu\_meta\_info 域为期望值，见备注 | | helper节点收到数据的 feishu\_meta\_info 域为期望值，见备注 | | 同预期结果一致 | | 通过 |
| 6 | 停止node-red测试环境 | |  |  | |  | |  | |  |
| **备注** | | 节点配置：  {  evalPropertyClass: 'group',  evalGroupProperty: 'open\_chat\_id',  evalUserProperty: 'open\_user\_id',  filterMode: 'regex',  filterValueList: [ '' ],  filterRegexPattern: '',  filterRegexIgnoreCase: false  }  输入数据1：  { payload: 'test\_message' }  输入数据2：  { payload: 'test\_message', feishu\_meta\_info: {} }  输入数据3：  { payload: 'test\_message', feishu\_meta\_info: { target: { user: [] } } }  输入数据4：  { payload: 'test\_message', feishu\_meta\_info: { target: { chat: [] } } }  feishu\_meta\_info 域期望值：  {  target: {  chat: [],  user: []  }  } | | | | | | | | |

* + 1. 飞书移除目标节点异常处理测试

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **测试用例名称** | | 飞书移除目标节点异常处理测试 | **测试用例标识** | | TC028 | | **测试需求标识** | | TR2.9 | |
| **简要描述** | | 测试飞书移除目标节点是否能在输入不符合预期的情况下正常运行报告相应错误 | | | | | | | | |
| **前提和约束** | | 无 | | | | | | | | |
| **测试方法** | | 单元测试 | | | | | | | | |
| **测试过程描述** | | | | | | | | | | |
| **序号** | **测试步骤** | | **输入** | **测试结果** | | | | **评价准则** | | **测试结论** |
| **预期结果** | | **实际结果** | |
| 1 | 搭建node-red测试环境 | |  |  | |  | |  | |  |
| 2 | 创建含有一个飞书移除目标节点节点和一个helper节点的工作流。 | | 飞书移除目标节点配置见备注。 |  | |  | |  | |  |
| 3 | 设置helper节点的消息处理回调函数用于后续判断 | |  |  | |  | |  | |  |
| 4 | 修改节点配置，部署工作流，输入数据并比对结果 | | 需修改配置：   * 筛选对象：“group” * 筛选属性：“open\_chat\_id”   输入数据1，见备注 | feishu\_meta\_info.target.chat的长度为1；feishu\_meta\_info.target.chat[0]为{}；  节点报告错误“Wrong chat info structure detected” | | feishu\_meta\_info.target.chat的长度为1；feishu\_meta\_info.target.chat[0]为{}；  节点报告错误“Wrong chat info structure detected” | | 同预期结果一致 | | 通过 |
| 需修改配置：   * 筛选对象：“group” * 筛选属性：“name”   输入数据2，见备注 | feishu\_meta\_info.target.chat的长度为1；feishu\_meta\_info.target.chat[0]为{}；  节点报告错误“Wrong chat info structure detected” | | feishu\_meta\_info.target.chat的长度为1；feishu\_meta\_info.target.chat[0]为{}；  节点报告错误“Wrong chat info structure detected” | | 同预期结果一致 | | 通过 |
| 需修改配置：   * 筛选对象：“user” * 筛选属性：“open\_user\_id”   输入数据3，见备注 | feishu\_meta\_info.target.user的长度为1；feishu\_meta\_info.target.user[0]为{}；  节点报告错误“Wrong user info structure detected” | | feishu\_meta\_info.target.user的长度为1；feishu\_meta\_info.target.user[0]为{}；  节点报告错误“Wrong user info structure detected” | | 同预期结果一致 | | 通过 |
| 5 | 停止node-red测试环境 | |  |  | |  | |  | |  |
| **备注** | | 节点配置：  {  evalPropertyClass: 'group',  evalGroupProperty: 'open\_chat\_id',  evalUserProperty: 'open\_user\_id',  filterMode: 'set',  filterValueList: [ 'test\_value' ],  filterRegexPattern: '',  filterRegexIgnoreCase: false  }  输入数据1：  {  payload: 'test\_message',  feishu\_meta\_info: {  target: { chat: [ {}, { chat\_id: 'test\_value' } ], user: [] }  }  }  输入数据2：  {  payload: 'test\_message',  feishu\_meta\_info: {  target: { chat: [ {}, { name: 'test\_value' } ], user: [] }  }  }  输入数据3：  {  payload: 'test\_message',  feishu\_meta\_info: { target: { chat: [], user: [ {}, 'test\_value' ] } }  } | | | | | | | | |

* + 1. 飞书移除目标节点配置异常处理测试

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **测试用例名称** | | 飞书移除目标节点配置异常处理测试 | **测试用例标识** | | TC029 | | **测试需求标识** | | TR2.9 | |
| **简要描述** | | 测试飞书移除目标节点是否能在配置不符合预期的情况下正常运行报告相应错误 | | | | | | | | |
| **前提和约束** | | 无 | | | | | | | | |
| **测试方法** | | 单元测试 | | | | | | | | |
| **测试过程描述** | | | | | | | | | | |
| **序号** | **测试步骤** | | **输入** | **测试结果** | | | | **评价准则** | | **测试结论** |
| **预期结果** | | **实际结果** | |
| 1 | 搭建node-red测试环境 | |  |  | |  | |  | |  |
| 2 | 创建含有一个飞书移除目标节点节点和一个helper节点的工作流。 | | 飞书移除目标节点配置见备注。 |  | |  | |  | |  |
| 3 | 设置helper节点的消息处理回调函数用于后续判断 | |  |  | |  | |  | |  |
| 4 | 修改节点配置，部署工作流，输入数据并比对结果 | | 需修改配置：   * 筛选模式：“invalid”   输入数据，见备注 | 节点报告错误“Unrecognized filterMode: invalid”；  helper节点收到数据的 feishu\_meta\_info 域为期望值，见备注 | | 节点报告错误“Unrecognized filterMode: invalid”；  helper节点收到数据的 feishu\_meta\_info 域为期望值，见备注 | | 同预期结果一致 | | 通过 |
| 需修改配置：   * 筛选对象：“invalid”   输入数据，见备注 | 节点报告错误“Unrecognized evalPropertyClass: invalid”；  helper节点收到数据的 feishu\_meta\_info 域为期望值，见备注 | | 节点报告错误“Unrecognized evalPropertyClass: invalid”；  helper节点收到数据的 feishu\_meta\_info 域为期望值，见备注 | | 同预期结果一致 | | 通过 |
| 需修改配置：   * 筛选对象：“group” * 筛选属性：“invalid”   输入数据，见备注 | 节点报告错误“Unrecognized evalGroupProperty: invalid”；  helper节点收到数据的 feishu\_meta\_info 域为期望值，见备注 | | 节点报告错误“Unrecognized evalGroupProperty: invalid”；  helper节点收到数据的 feishu\_meta\_info 域为期望值，见备注 | | 同预期结果一致 | | 通过 |
| 需修改配置：   * 筛选对象：“user” * 筛选属性：“invalid”   输入数据，见备注 | 节点报告错误“Unrecognized evalUserProperty: invalid”；  helper节点收到数据的 feishu\_meta\_info 域为期望值，见备注 | | 节点报告错误“Unrecognized evalUserProperty: invalid”；  helper节点收到数据的 feishu\_meta\_info 域为期望值，见备注 | | 同预期结果一致 | | 通过 |
| 5 | 停止node-red测试环境 | |  |  | |  | |  | |  |
| **备注** | | 节点配置：  {  evalPropertyClass: "group",  evalGroupProperty: "open\_chat\_id",  evalUserProperty: "open\_user\_id",  filterMode: "set",  filterValueList: ["test\_value"],  filterRegexPattern: "",  filterRegexIgnoreCase: false  }  输入数据：  {  payload: 'test\_message',  feishu\_meta\_info: { target: { chat: [ 'test\_chat\_id' ], user: [ 'test\_user\_id' ] } }  }  feishu\_meta\_info 域期望值：  { target: { chat: [ 'test\_chat\_id' ], user: [ 'test\_user\_id' ] } | | | | | | | | |

## 创建飞书发送消息节点测试

* + 1. 飞书发送消息节点加载测试

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **测试用例名称** | | feishu-send  节点加载测试 | **测试用例标识** | | TC030 | | **测试需求标识** | | TR2.10 | |
| **简要描述** | | 测试 feishu-send节点是否能正确加载 | | | | | | | | |
| **前提和约束** | | 无 | | | | | | | | |
| **测试方法** | | 冒烟测试 | | | | | | | | |
| **测试过程描述** | | | | | | | | | | |
| **序号** | **测试步骤** | | **输入** | **测试结果** | | | | **评价准则** | | **测试结论** |
| **预期结果** | | **实际结果** | |
| 1 | 搭建node-red测试环境 | |  |  | |  | |  | |  |
| 2 | 创建含有一个feishu-send节点和一个feishu-config节点的工作流 | |  |  | |  | |  | |  |
| 3 | 按照飞书tenant\_access\_token API 架设nock拦截器 | |  |  | |  | |  | |  |
| 4 | 部署工作流 | |  |  | |  | |  | |  |
| 5 | 检查节点加载状态 | |  | feishu-config节点获得tenant\_token | | feishu-config节点获得tenant\_token | | 检查tenant\_token是否与预设值相同 | | 通过 |
|  | feishu-send节点没有报错 | | feishu-send节点没有报错 | | 检查是否有未处理的异常 | | 通过 |
| 6 | 停止node-red测试环境 | |  |  | |  | |  | |  |
| **备注** | | 无 | | | | | | | | |

* + 1. 飞书发送消息节点发送群聊图片测试

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **测试用例名称** | | feishu-send  节点发送群聊图片测试 | **测试用例标识** | | TC031 | | **测试需求标识** | | TR2.10 | |
| **简要描述** | | 测试 feishu-send节点是否能正确发送群聊图片 | | | | | | | | |
| **前提和约束** | | 无 | | | | | | | | |
| **测试方法** | | 单元测试 | | | | | | | | |
| **测试过程描述** | | | | | | | | | | |
| **序号** | **测试步骤** | | **输入** | **测试结果** | | | | **评价准则** | | **测试结论** |
| **预期结果** | | **实际结果** | |
| 1 | 搭建node-red测试环境 | |  |  | |  | |  | |  |
| 2 | 创建含有一个feishu-config,feishu-send, 节点的工作流 | |  |  | |  | |  | |  |
| 3 | 按照飞书tenant\_access\_token API 架设nock拦截器 | |  |  | |  | |  | |  |
| 4 | 按照飞书发送消息API架设nock拦截器 | |  |  | |  | |  | |  |
| 5 | 部署工作流 | |  |  | |  | |  | |  |
| 6 | 检查节点工作状态 | |  | feishu-config 获得tenant\_token | | feishu-config 获得tenant\_token | | 检查是否与预设的token一致 | | 通过 |
| 测试消息1 | 拦截器收到feishu-send的请求，请求格式正确 | | 拦截器收到feishu-send的请求，请求格式正确 | | 检查拦截到的请求是否符合预设的格式 | | 通过 |
| 7 | 停止node-red测试环境 | |  |  | |  | |  | |  |
| **备注** | | 测试消息1  {  payload: {  msg: {  image\_key: TEST\_IMAGE\_KEY  }  },  feishu\_meta\_info: {  target: {  chat: TEST\_GROUP\_LIST,  user: []  },  chat: [],  user: []  }  }  预设请求格式  {  chat\_id: TEST\_GROUP\_LIST[i],  msg\_type: ‘image’,  content: {  image\_key: TEST\_IMAGE\_KEY  }  } | | | | | | | | |

* + 1. 飞书发送消息节点发送群聊文本测试

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **测试用例名称** | | feishu-send  节点发送群聊文本测试 | **测试用例标识** | | TC032 | | **测试需求标识** | | TR2.10 | |
| **简要描述** | | 测试 feishu-send节点是否能正确发送群聊文本 | | | | | | | | |
| **前提和约束** | | 无 | | | | | | | | |
| **测试方法** | | 单元测试 | | | | | | | | |
| **测试过程描述** | | | | | | | | | | |
| **序号** | **测试步骤** | | **输入** | **测试结果** | | | | **评价准则** | | **测试结论** |
| **预期结果** | | **实际结果** | |
| 1 | 搭建node-red测试环境 | |  |  | |  | |  | |  |
| 2 | 创建含有一个feishu-config,feishu-send, 节点的工作流 | |  |  | |  | |  | |  |
| 3 | 按照飞书tenant\_access\_token API 架设nock拦截器 | |  |  | |  | |  | |  |
| 4 | 按照飞书发送消息API架设nock拦截器 | |  |  | |  | |  | |  |
| 5 | 部署工作流 | |  |  | |  | |  | |  |
| 6 | 检查节点加载状态 | |  | feishu-config 获得tenant\_token | | feishu-config 获得tenant\_token | | 检查是否与预设的token一致 | | 通过 |
| 测试消息1 | 拦截器收到feishu-send的请求，请求格式正确 | | 拦截器收到feishu-send的请求，请求格式正确 | | 检查拦截到的请求是否符合预设的格式 | | 通过 |
| 7 | 停止node-red测试环境 | |  |  | |  | |  | |  |
| **备注** | | 测试消息1  {  payload: {  msg: {  text: TEST\_TEXT  }  },  feishu\_meta\_info: {  target: {  chat: TEST\_GROUP\_LIST,  user: []  },  chat: [],  user: []  }  }  预设请求格式  {  chat\_id: TEST\_GROUP\_LIST[i],  msg\_type: ‘text’,  content: {  text: TEST\_TEXT  }  } | | | | | | | | |

* + 1. 飞书发送消息节点发送私聊图片测试

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **测试用例名称** | | feishu-send  节点发送私聊图片测试 | **测试用例标识** | | TC033 | | **测试需求标识** | | TR2.10 | |
| **简要描述** | | 测试 feishu-send节点是否能正确发送私聊图片 | | | | | | | | |
| **前提和约束** | | 无 | | | | | | | | |
| **测试方法** | | 单元测试 | | | | | | | | |
| **测试过程描述** | | | | | | | | | | |
| **序号** | **测试步骤** | | **输入** | **测试结果** | | | | **评价准则** | | **测试结论** |
| **预期结果** | | **实际结果** | |
| 1 | 搭建node-red测试环境 | |  |  | |  | |  | |  |
| 2 | 创建含有一个feishu-config,feishu-send, 节点的工作流 | |  |  | |  | |  | |  |
| 3 | 按照飞书tenant\_access\_token API 架设nock拦截器 | |  |  | |  | |  | |  |
| 4 | 按照飞书发送消息API架设nock拦截器 | |  |  | |  | |  | |  |
| 5 | 部署工作流 | |  |  | |  | |  | |  |
| 6 | 检查节点工作状态 | |  | feishu-config 获得tenant\_token | | feishu-config 获得tenant\_token | | 检查是否与预设的token一致 | | 通过 |
| 测试消息1 | 拦截器收到feishu-send的请求，请求格式正确 | | 拦截器收到feishu-send的请求，请求格式正确 | | 检查拦截到的请求是否符合预设的格式 | | 通过 |
| 7 | 停止node-red测试环境 | |  |  | |  | |  | |  |
| **备注** | | 测试消息1  {  payload: {  msg: {  image\_key: TEST\_IMAGE\_KEY  }  },  feishu\_meta\_info: {  target: {  chat:[],  user: TEST\_USER\_LIST  },  chat: [],  user: []  }  }  预设请求格式  {  open\_id: TEST\_USER\_LIST[i],  msg\_type: ‘image’,  content: {  image\_key: TEST\_IMAGE\_KEY  }  } | | | | | | | | |

* + 1. 飞书发送消息节点发送私聊文本测试

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **测试用例名称** | | feishu-send  节点发送私聊文本测试 | **测试用例标识** | | TC034 | | **测试需求标识** | | TR2.10 | |
| **简要描述** | | 测试 feishu-send节点是否能正确发送私聊文本 | | | | | | | | |
| **前提和约束** | | 无 | | | | | | | | |
| **测试方法** | | 单元测试 | | | | | | | | |
| **测试过程描述** | | | | | | | | | | |
| **序号** | **测试步骤** | | **输入** | **测试结果** | | | | **评价准则** | | **测试结论** |
| **预期结果** | | **实际结果** | |
| 1 | 搭建node-red测试环境 | |  |  | |  | |  | |  |
| 2 | 创建含有一个feishu-config,feishu-send, 节点的工作流 | |  |  | |  | |  | |  |
| 3 | 按照飞书tenant\_access\_token API 架设nock拦截器 | |  |  | |  | |  | |  |
| 4 | 按照飞书发送消息API架设nock拦截器 | |  |  | |  | |  | |  |
| 5 | 部署工作流 | |  |  | |  | |  | |  |
| 6 | 检查节点工作状态 | |  | feishu-config 获得tenant\_token | | feishu-config 获得tenant\_token | | 检查是否与预设的token一致 | | 通过 |
| 测试消息1 | 拦截器收到feishu-send的请求，请求格式正确 | | 拦截器收到feishu-send的请求，请求格式正确 | | 检查拦截到的请求是否符合预设的格式 | | 通过 |
| 7 | 停止node-red测试环境 | |  |  | |  | |  | |  |
| **备注** | | 测试消息1  {  payload: {  msg: {  text: TEST\_TEXT  }  },  feishu\_meta\_info: {  target: {  chat:[],  user: TEST\_USER\_LIST  },  chat: [],  user: []  }  }  预设请求格式  {  open\_id: TEST\_USER\_LIST[i],  msg\_type: ‘text’,  content: {  text: TEST\_TEXT  }  } | | | | | | | | |

## 创建图表渲染节点测试

* + 1. 图表渲染节点部署测试

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **测试用例名称** | | chart-render节点部署测试 | **测试用例标识** | | TC035 | | **测试需求标识** | | TR2.11 | |
| **简要描述** | | 测试chart-render节点能否正常部署 | | | | | | | | |
| **前提和约束** | | 无 | | | | | | | | |
| **测试方法** | | 冒烟测试 | | | | | | | | |
| **测试过程描述** | | | | | | | | | | |
| **序号** | **测试步骤** | | **输入** | **测试结果** | | | | **评价准则** | | **测试结论** |
| **预期结果** | | **实际结果** | |
| 1 | 搭建node-red测试环境 | |  |  | |  | |  | |  |
| 2 | 创建一个含有chart-render节点的工作流，chart-render节点的name属性为test | |  |  | |  | |  | |  |
| 3 | 部署工作流 | |  |  | |  | |  | |  |
| 4 | 获取chart-render节点的name属性 | |  | 获取到name属性值为test | | 获取到name属性值为test | | 实际结果与预期结果内容一致 | | 通过 |
| 5 | 停止node-red测试环境 | |  |  | |  | |  | |  |
| **备注** | | 无 | | | | | | | | |

* + 1. 图表渲染节点Line chart绘制测试

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **测试用例名称** | | chart-render节点Line chart绘制测试 | **测试用例标识** | | TC036 | | **测试需求标识** | | TR2.11 | |
| **简要描述** | | 测试chart-render节点是否能正确绘制Line chart | | | | | | | | |
| **前提和约束** | | 无 | | | | | | | | |
| **测试方法** | | 单元测试 | | | | | | | | |
| **测试过程描述** | | | | | | | | | | |
| **序号** | **测试步骤** | | **输入** | **测试结果** | | | | **评价准则** | | **测试结论** |
| **预期结果** | | **实际结果** | |
| 1 | 搭建node-red测试环境 | |  |  | |  | |  | |  |
| 2 | 创建含有一个chart-render窗口节点和一个helper节点的工作流，chart-render的chartType参数为Line chart，Output参数为base64，其他为空 | |  |  | |  | |  | |  |
| 3 | 部署工作流 | |  |  | |  | |  | |  |
| 4 | 设置helper节点的消息处理回调函数用于后续判断 | |  |  | |  | |  | |  |
| 5 | 向chart-render节点输入数据 | | 输入number类型的消息：  {payload: 123} | helper节点收到数据保存到output/${TESTID}\_6.1.jpg | | helper节点收到数据保存到output/${TESTID}\_6.1.jpg | | 与gt文件夹中的同名图片肉眼比对一致 | | 通过 |
| 依次输入两个number类型的消息：1.{payload: 123}  2.{payload:123} | helper节点收到两次数据，将保存到output/${TESTID}\_6.2.1.jpg和output/${TESTID}\_6.2.2.jpg | | helper节点收到两次数据，将保存到output/${TESTID}\_6.2.1.jpg和output/${TESTID}\_6.2.2.jpg | | 与gt文件夹中的同名图片肉眼比对一致 | | 通过 |
| 依次输入两个Array类型的消息：1.{payload: [1, 2, 3]}  2.{payload: [1, 2, 3]} | helper节点收到两次数据，将保存到output、${TESTID}\_6.3.1.jpg和output/${TESTID}\_6.3.2.jpg | | helper节点收到两次数据，将保存到output、${TESTID}\_6.3.1.jpg和output/${TESTID}\_6.3.2.jpg | | 与gt文件夹中的同名图片肉眼比对一致 | | 通过 |
| 输入非法类型的消息（以bool类型为例）：  {payload: false} | chart-render节点抛出异常被node-red框架捕获，helper节点不会收到消息 | | chart-render节点抛出异常被node-red框架捕获，helper节点不会收到消息 | | chart-render节点没有输出 | | 通过 |
| 输入带横坐标的number信息（并包含横坐标格式化字符串）：  {payload: "null", x: "2020-10-1", y: 123, format: "date:%Y-%m-%d"} | helper节点收到数据保存到output/${TESTID}\_6.4.jpg | | helper节点收到数据保存到output/${TESTID}\_6.4.jpg | | 与gt文件夹中的同名图片肉眼比对一致 | | 通过 |
| 输入带横坐标的Array信息（不包含横坐标格式化字符串）：  {payload: "null", x: ["x1", "x2"], y: [123, 123]} | helper节点收到数据保存到output/${TESTID}\_6.5.jpg | | helper节点收到数据保存到output/${TESTID}\_6.5.jpg | | 与gt文件夹中的同名图片肉眼比对一致 | | 通过 |
| 输入带横坐标的错误格式信息：  {payload: "null", x: false, y: false} | chart-render节点抛出异常被node-red框架捕获，helper节点不会收到消息 | | chart-render节点抛出异常被node-red框架捕获，helper节点不会收到消息 | | chart-render节点没有输出 | | 通过 |
| 输入带config的信息，以修改节点的配置：  { payload: 123, config: { chartType: "Line chart", title: "", xLabel: "", yLabel: "", dataWindow: 0, output: "base64", height: 200, width: 400}} | helper节点收到数据保存到output/${TESTID}\_6.7.jpg | | helper节点收到数据保存到output/${TESTID}\_6.7.jpg | | 与gt文件夹中的同名图片肉眼比对一致 | | 通过 |
| 输入带config的信息，但是config内容是空的：  { payload: 123, config: {}} | helper节点收到数据保存到output/${TESTID}\_6.8.jpg | | helper节点收到数据保存到output/${TESTID}\_6.8.jpg | | 与gt文件夹中的同名图片肉眼比对一致 | | 通过 |
| 6 | 停止node-red测试环境 | |  |  | |  | |  | |  |
| **备注** | |  | | | | | | | | |

* + 1. 图表渲染节点Bar chart绘制测试

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **测试用例名称** | | chart-render节点Bar chart绘制测试 | **测试用例标识** | | TC037 | | **测试需求标识** | | TR2.11 | |
| **简要描述** | | 测试chart-render节点是否能正确绘制Bar chart | | | | | | | | |
| **前提和约束** | | 无 | | | | | | | | |
| **测试方法** | | 单元测试 | | | | | | | | |
| **测试过程描述** | | | | | | | | | | |
| **序号** | **测试步骤** | | **输入** | **测试结果** | | | | **评价准则** | | **测试结论** |
| **预期结果** | | **实际结果** | |
| 1 | 搭建node-red测试环境 | |  |  | |  | |  | |  |
| 2 | 创建含有一个chart-render窗口节点和一个helper节点的工作流，chart-render的chartType参数为Bar chart，Output参数为base64，其他为空 | |  |  | |  | |  | |  |
| 3 | 部署工作流 | |  |  | |  | |  | |  |
| 4 | 设置helper节点的消息处理回调函数用于后续判断 | |  |  | |  | |  | |  |
| 5 | 向chart-render节点输入数据 | | 输入消息：  {payload: 123} | helper节点收到数据{payload: <base64-value-${TEXTID}\_5.1>} | | helper节点收到数据{payload: <base64-value-${TEXTID}\_5.1>} | | 与gt文件夹中的同名图片肉眼比对一致 | | 通过 |
| 输入带横坐标的信息（并包含横坐标格式化字符串）：  {payload: "null", x: "2020-10-1", y: 123, format: "date:%Y-%m-%d"} | helper节点收到数据保存到output/${TESTID}\_5.2.jpg | | helper节点收到数据保存到output/${TESTID}\_5.2.jpg | | 与gt文件夹中的同名图片肉眼比对一致 | | 通过 |
| 6 | 停止node-red测试环境 | |  |  | |  | |  | |  |
| **备注** | |  | | | | | | | | |

* + 1. 图表渲染节点Pie chart绘制测试

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **测试用例名称** | | chart-render节点Pie chart绘制测试 | **测试用例标识** | | TC038 | | **测试需求标识** | | TR2.11 | |
| **简要描述** | | 测试chart-render节点是否能正确绘制Pie chart | | | | | | | | |
| **前提和约束** | | 无 | | | | | | | | |
| **测试方法** | | 单元测试 | | | | | | | | |
| **测试过程描述** | | | | | | | | | | |
| **序号** | **测试步骤** | | **输入** | **测试结果** | | | | **评价准则** | | **测试结论** |
| **预期结果** | | **实际结果** | |
| 1 | 搭建node-red测试环境 | |  |  | |  | |  | |  |
| 2 | 创建含有一个chart-render窗口节点和一个helper节点的工作流，chart-render的chartType参数为Pie chart，Output参数为base64，其他为空 | |  |  | |  | |  | |  |
| 3 | 部署工作流 | |  |  | |  | |  | |  |
| 4 | 设置helper节点的消息处理回调函数用于后续判断 | |  |  | |  | |  | |  |
| 5 | 向chart-render节点输入数据 | | 输入消息：  {payload: 123} | helper节点收到数据保存到output/${TESTID}\_5.1.jpg | | helper节点收到数据保存到output/${TESTID}\_5.1.jpg | | 与gt文件夹中的同名图片肉眼比对一致 | | 通过 |
| 输入带横坐标的信息（并包含横坐标格式化字符串）：  {payload: "null", x: "2020-10-1", y: 123, format: "date:%Y-%m-%d"} | helper节点收到数据保存到output/${TESTID}\_5.2.jpg | | helper节点收到数据保存到output/${TESTID}\_5.2.jpg | | 与gt文件夹中的同名图片肉眼比对一致 | | 通过 |
| 6 | 停止node-red测试环境 | |  |  | |  | |  | |  |
| **备注** | |  | | | | | | | | |

* + 1. 图表渲染节点Radar chart绘制测试

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **测试用例名称** | | chart-render节点Radar chart绘制测试 | **测试用例标识** | | TC039 | | **测试需求标识** | | TR2.11 | |
| **简要描述** | | 测试chart-render节点是否能正确绘制Radar chart | | | | | | | | |
| **前提和约束** | | 无 | | | | | | | | |
| **测试方法** | | 单元测试 | | | | | | | | |
| **测试过程描述** | | | | | | | | | | |
| **序号** | **测试步骤** | | **输入** | **测试结果** | | | | **评价准则** | | **测试结论** |
| **预期结果** | | **实际结果** | |
| 1 | 搭建node-red测试环境 | |  |  | |  | |  | |  |
| 2 | 创建含有一个chart-render窗口节点和一个helper节点的工作流，chart-render的chartType参数为Radar chart，Output参数为base64，其他为空 | |  |  | |  | |  | |  |
| 3 | 部署工作流 | |  |  | |  | |  | |  |
| 4 | 设置helper节点的消息处理回调函数用于后续判断 | |  |  | |  | |  | |  |
| 5 | 向chart-render节点输入数据 | | 输入消息：  {payload: 123} | helper节点收到数据保存到output/${TESTID}\_5.1,jpg | | helper节点收到数据保存到output/${TESTID}\_5.1,jpg | | 与gt文件夹中的同名图片肉眼比对一致 | | 通过 |
| 输入带横坐标的信息（并包含横坐标格式化字符串）：  {payload: "null", x: "2020-10-1", y: 123, format: "date:%Y-%m-%d"} | helper节点收到数据保存到output/${TESTID}\_5.2.jpg | | helper节点收到数据保存到output/${TESTID}\_5.2.jpg | | 与gt文件夹中的同名图片肉眼比对一致 | | 通过 |
| 6 | 停止node-red测试环境 | |  |  | |  | |  | |  |
| **备注** | |  | | | | | | | | |

* + 1. 图表渲染节点Polar area chart绘制测试

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **测试用例名称** | | chart-render节点Polar area chart绘制测试 | **测试用例标识** | | TC040 | | **测试需求标识** | | TR2.11 | |
| **简要描述** | | 测试chart-render节点是否能正确绘制Polar area chart | | | | | | | | |
| **前提和约束** | | 无 | | | | | | | | |
| **测试方法** | | 单元测试 | | | | | | | | |
| **测试过程描述** | | | | | | | | | | |
| **序号** | **测试步骤** | | **输入** | **测试结果** | | | | **评价准则** | | **测试结论** |
| **预期结果** | | **实际结果** | |
| 1 | 搭建node-red测试环境 | |  |  | |  | |  | |  |
| 2 | 创建含有一个chart-render窗口节点和一个helper节点的工作流，chart-render的chartType参数为Polar area chart，Output参数为base64，其他为空 | |  |  | |  | |  | |  |
| 3 | 部署工作流 | |  |  | |  | |  | |  |
| 4 | 设置helper节点的消息处理回调函数用于后续判断 | |  |  | |  | |  | |  |
| 5 | 向chart-render节点输入数据 | | 输入消息：  {payload: 123} | helper节点收到数据，保存到output/${TESTID}\_5.1.jpg | | helper节点收到数据，保存到output/${TESTID}\_5.1.jpg | | 与gt文件夹中的同名图片肉眼比对一致 | | 通过 |
| 输入带横坐标的信息（并包含横坐标格式化字符串）：  {payload: "null", x: "2020-10-1", y: 123, format: "date:%Y-%m-%d"} | helper节点收到数据，保存到output/${TESTID}\_5.2.jpg | | helper节点收到数据，保存到output/${TESTID}\_5.2.jpg | | 与gt文件夹中的同名图片肉眼比对一致 | | 通过 |
| 6 | 停止node-red测试环境 | |  |  | |  | |  | |  |
| **备注** | | 由于base64字符串很长，将<base64-value-filename>的值以文件形式存储在./filename.chart-render.base64.test中 | | | | | | | | |

* + 1. 图表渲染节点修改图表长宽测试

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **测试用例名称** | | chart-render节点修改图表长宽测试 | **测试用例标识** | | TC041 | | **测试需求标识** | | TR2.11 | |
| **简要描述** | | 测试chart-render节点是否能正确绘制给定尺寸的图表 | | | | | | | | |
| **前提和约束** | | 无 | | | | | | | | |
| **测试方法** | | 单元测试 | | | | | | | | |
| **测试过程描述** | | | | | | | | | | |
| **序号** | **测试步骤** | | **输入** | **测试结果** | | | | **评价准则** | | **测试结论** |
| **预期结果** | | **实际结果** | |
| 1 | 搭建node-red测试环境 | |  |  | |  | |  | |  |
| 2 | 创建含有一个chart-render窗口节点和一个helper节点的工作流，chart-render的chartType参数为Line chart，Output参数为base64，Height参数为200，Width参数为400，其他为空 | |  |  | |  | |  | |  |
| 3 | 部署工作流 | |  |  | |  | |  | |  |
| 4 | 设置helper节点的消息处理回调函数用于后续判断 | |  |  | |  | |  | |  |
| 5 | 向chart-render节点输入数据 | | 输入消息：  {payload: 123} | helper节点收到数据，保存到output/${TESTID}\_5.jpg | | helper节点收到数据，保存到output/${TESTID}\_5.jpg | | 与gt文件夹中的同名图片肉眼比对一致 | | 通过 |
| 6 | 停止node-red测试环境 | |  |  | |  | |  | |  |
| **备注** | |  | | | | | | | | |

* + 1. 图表渲染节点数据窗口功能测试

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **测试用例名称** | | chart-render节点数据窗口测试 | **测试用例标识** | | TC042 | | **测试需求标识** | | TR2.11 | |
| **简要描述** | | 测试chart-render节点的数据窗口功能是否能正常工作 | | | | | | | | |
| **前提和约束** | | 无 | | | | | | | | |
| **测试方法** | | 单元测试 | | | | | | | | |
| **测试过程描述** | | | | | | | | | | |
| **序号** | **测试步骤** | | **输入** | **测试结果** | | | | **评价准则** | | **测试结论** |
| **预期结果** | | **实际结果** | |
| 1 | 搭建node-red测试环境 | |  |  | |  | |  | |  |
| 2 | 创建含有一个chart-render窗口节点和一个helper节点的工作流，chart-render的chartType参数为Bar chart，Output参数为base64，DataWindow参数为1，其他为空 | |  |  | |  | |  | |  |
| 3 | 部署工作流 | |  |  | |  | |  | |  |
| 4 | 设置helper节点的消息处理回调函数用于后续判断 | |  |  | |  | |  | |  |
| 5 | 向chart-render节点输入数据 | | 依次输入两个消息：1.{payload: 123}  2.{payload:123} | helper节点收到两次base64 串，保存到output/${TESTID}\_5.1.1.jpg和output/${TESTID}\_5.1.2.jpg | | helper节点收到两次base64 串，保存到output/${TESTID}\_5.1.1.jpg和output/${TESTID}\_5.1.2.jpg | | 与gt文件夹中的同名图片肉眼比对一致 | | 通过 |
| 6 | 停止node-red测试环境 | |  |  | |  | |  | |  |
| **备注** | |  | | | | | | | | |

* + 1. 图表渲染节点binary输出测试

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **测试用例名称** | | chart-render节点binary输出测试 | **测试用例标识** | | TC043 | | **测试需求标识** | | TR2.11 | |
| **简要描述** | | 测试chart-render节点是否能正常输出binary格式的图表 | | | | | | | | |
| **前提和约束** | | 无 | | | | | | | | |
| **测试方法** | | 单元测试 | | | | | | | | |
| **测试过程描述** | | | | | | | | | | |
| **序号** | **测试步骤** | | **输入** | **测试结果** | | | | **评价准则** | | **测试结论** |
| **预期结果** | | **实际结果** | |
| 1 | 搭建node-red测试环境 | |  |  | |  | |  | |  |
| 2 | 创建含有一个chart-render窗口节点和一个helper节点的工作流，chart-render的chartType参数为Line chart，Output参数为binary，其他为空 | |  |  | |  | |  | |  |
| 3 | 部署工作流 | |  |  | |  | |  | |  |
| 4 | 设置helper节点的消息处理回调函数用于后续判断 | |  |  | |  | |  | |  |
| 5 | 向chart-render节点输入数据 | | 输入消息：{payload: 123} | helper节点收到一个binary串，存储图片到output/${TESTID}\_5.jpg | | helper节点收到一个binary串，存储图片到output/${TESTID}\_5.jpg | | 与gt文件夹中的同名图片肉眼比对一致 | | 通过 |
| 6 | 停止node-red测试环境 | |  |  | |  | |  | |  |
| **备注** | |  | | | | | | | | |

## 创建模板渲染节点测试

* + 1. 模板渲染节点部署测试

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **测试用例名称** | | 模板渲染节点部署测试 | **测试用例标识** | | TC044 | | **测试需求标识** | | TR2.12 | |
| **简要描述** | | 测试模板渲染节点是否能够正确部署 | | | | | | | | |
| **前提和约束** | | 无 | | | | | | | | |
| **测试方法** | | 冒烟测试 | | | | | | | | |
| **测试过程描述** | | | | | | | | | | |
| **序号** | **测试步骤** | | **输入** | **测试结果** | | | | **评价准则** | | **测试结论** |
| **预期结果** | | **实际结果** | |
| 1 | 建立node-red测试环境 | |  |  | |  | |  | |  |
| 2 | 创建含有一个name属性为"template-render"，其他属性不配置的模板渲染节点的工作流 | |  |  | |  | |  | |  |
| 3 | 部署工作流 | |  |  | |  | |  | |  |
| 4 | 获取模板渲染节点的name属性 | |  | 属性值为"template-render" | | 属性值为"template-render" | | 属性值一致 | | 通过 |
| 5 | 停止node-red测试环境 | |  |  | |  | |  | |  |
| **备注** | | 无 | | | | | | | | |

* + 1. 模板渲染节点功能正确性测试

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **测试用例名称** | | 模板渲染节点功能正确性测试 | **测试用例标识** | | TC045 | | **测试需求标识** | | TR2.12 | |
| **简要描述** | | 测试模板渲染节点是否能够依据输入正确渲染模板 | | | | | | | | |
| **前提和约束** | | 无 | | | | | | | | |
| **测试方法** | | 单元测试 | | | | | | | | |
| **测试过程描述** | | | | | | | | | | |
| **序号** | **测试步骤** | | **输入** | **测试结果** | | | | **评价准则** | | **测试结论** |
| **预期结果** | | **实际结果** | |
| 1 | 建立node-red测试环境 | |  |  | |  | |  | |  |
| 2 | 创建含有一个模板渲染节点和一个helper节点的工作流， 配置leftDelimiter:"{{",rightDelimiter:"}}",clearWhenOutput:"true",outputWhenAllSet:"true",template:"#Header\n##Header2\nToday is {{date}}\nThis is image ![]({{image}})", | |  |  | |  | |  | |  |
| 3 | 部署工作流 | |  |  | |  | |  | |  |
| 4 | 设置helper节点的消息处理回调函数用于后续判断 | |  |  | |  | |  | |  |
| 5 | 向模板渲染节点输入数据 | | {payload:{date:"May 13"}} |  | |  | |  | |  |
| 6 | 向模板渲染节点输入数据 | | {payload:{image:" https://test.com/img "}} | helper收到消息，payload为"#Header\n##Header2\nToday is May 13\nThis is image ![](https://test.com/img) " | | helper收到消息，payload为"#Header\n##Header2\nToday is May 13\nThis is image ![](https://test.com/img) " | | 只有在本次输入才有输出，且数据一致 | | 通过 |
| 7 | 停止node-red测试环境 | |  |  | |  | |  | |  |
| **备注** | | 无 | | | | | | | | |

* + 1. 模板渲染节点异常处理测试

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **测试用例名称** | | 模板渲染节点异常处理测试 | **测试用例标识** | | TC046 | | **测试需求标识** | | TR2.12 | |
| **简要描述** | | 测试模板渲染节点对于异常输入是否能够报错 | | | | | | | | |
| **前提和约束** | | 无 | | | | | | | | |
| **测试方法** | | 单元测试 | | | | | | | | |
| **测试过程描述** | | | | | | | | | | |
| **序号** | **测试步骤** | | **输入** | **测试结果** | | | | **评价准则** | | **测试结论** |
| **预期结果** | | **实际结果** | |
| 1 | 建立node-red测试环境 | |  |  | |  | |  | |  |
| 2 | 创建含有一个模板渲染节点和一个helper节点的工作流， 配置leftDelimiter:"{{",rightDelimiter:"}}",clearWhenOutput:"true",outputWhenAllSet:"true",template:"#Header\n##Header2\nToday is {{date}}\nThis is image ![]({{image}})", | |  |  | |  | |  | |  |
| 3 | 部署工作流 | |  |  | |  | |  | |  |
| 4 | 向模板渲染节点输入数据 | | {payload:{date:"May 13"}} |  | |  | |  | |  |
| 5 | 向模板渲染节点输入数据 | | {} | 模板渲染节点报错，错误信息开头为"Cannot convert undefined" | | 模板渲染节点报错，错误信息开头为"Cannot convert undefined" | | 报错，且报错信息一致 | | 通过 |
| 6 | 停止node-red测试环境 | |  |  | |  | |  | |  |
| **备注** | | 无 | | | | | | | | |

* + 1. 模板渲染节点特殊输入处理测试

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **测试用例名称** | | 模板渲染节点忽略空值输入测试 | **测试用例标识** | | TC047 | | **测试需求标识** | | TR2.12 | |
| **简要描述** | | 测试模板渲染节点是否能够忽略输入中的空值，避免非预期的输出 | | | | | | | | |
| **前提和约束** | | 无 | | | | | | | | |
| **测试方法** | | 单元测试 | | | | | | | | |
| **测试过程描述** | | | | | | | | | | |
| **序号** | **测试步骤** | | **输入** | **测试结果** | | | | **评价准则** | | **测试结论** |
| **预期结果** | | **实际结果** | |
| 1 | 建立node-red测试环境 | |  |  | |  | |  | |  |
| 2 | 创建含有一个模板渲染节点和一个helper节点的工作流，配置leftDelimiter:"{{",rightDelimiter:"}}",clearWhenOutput:"true",outputWhenAllSet:"true",template:"#Header\n##Header2\nToday is {{date}}\nThis is image ![]({{image}})", | |  |  | |  | |  | |  |
| 3 | 部署工作流 | |  |  | |  | |  | |  |
| 4 | 设置helper节点的消息处理回调函数用于后续判断 | |  |  | |  | |  | |  |
| 5 | 向模板渲染节点输入数据 | | {payload:{date:"May 13"}} |  | |  | |  | |  |
| 6 | 向模板渲染节点输入数据 | | {payload:{date:null}} |  | |  | |  | |  |
| 7 | 向模板渲染节点输入数据 | | {payload:{image:"https://test.com/img"}} | helper收到消息，payload为"#Header\n##Header2\nToday is May 13\nThis is image ![](https://test.com/img) " | | helper收到消息，payload为"#Header\n##Header2\nToday is May 13\nThis is image ![](https://test.com/img) " | | 只有在本次输入才有输出，且数据一致 | | 通过 |
| 8 | 停止node-red测试环境 | |  |  | |  | |  | |  |
| **备注** | | 无 | | | | | | | | |

* + 1. 模板渲染节点输出时清空字段功能正确性测试

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **测试用例名称** | | 模板渲染节点输出时清空字段功能正确性测试 | **测试用例标识** | | TC048 | | **测试需求标识** | | TR2.12 | |
| **简要描述** | | 测试模板渲染节点是否能够在输出时清空所有字段，以便渲染全新的模板 | | | | | | | | |
| **前提和约束** | | 无 | | | | | | | | |
| **测试方法** | | 单元测试 | | | | | | | | |
| **测试过程描述** | | | | | | | | | | |
| **序号** | **测试步骤** | | **输入** | **测试结果** | | | | **评价准则** | | **测试结论** |
| **预期结果** | | **实际结果** | |
| 1 | 建立node-red测试环境 | |  |  | |  | |  | |  |
| 2 | 创建含有一个模板渲染节点和一个helper节点的工作流，配置leftDelimiter:"{{",rightDelimiter:"}}",clearWhenOutput:"true",outputWhenAllSet:"true",template:"#Header\n##Header2\nToday is {{date}}\nThis is image ![]({{image}})", | |  |  | |  | |  | |  |
| 3 | 部署工作流 | |  |  | |  | |  | |  |
| 4 | 设置helper节点的消息处理回调函数用于后续判断 | |  |  | |  | |  | |  |
| 5 | 向模板渲染节点输入数据 | | {payload:{date:"May 13"}} |  | |  | |  | |  |
| 6 | 向模板渲染节点输入数据 | | {payload:{image:"https://test.com/img"}} | helper收到消息，payload为"#Header\n##Header2\nToday is May 13\nThis is image ![](https://test.com/img) " | | helper收到消息，payload为"#Header\n##Header2\nToday is May 13\nThis is image ![](https://test.com/img) " | | 本次输入有输出，且数据一致 | | 通过 |
| 7 | 向模板渲染节点输入数据 | | {payload:{date:"May 14"}} |  | |  | |  | |  |
| 8 | 向模板渲染节点输入数据 | | {payload:{image:"https://test2.com/img"}} | helper收到消息，payload为"#Header\n##Header2\nToday is May 14\nThis is image ![](https://test2.com/img) " | | helper收到消息，payload为"#Header\n##Header2\nToday is May 14\nThis is image ![](https://test2.com/img) " | | 本次输有输出，且数据一致 | | 通过 |
| 9 | 停止node-red测试环境 | |  |  | |  | |  | |  |
| **备注** | | 无 | | | | | | | | |

* + 1. 模板渲染节点输出后不清空字段功能正确性测试

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **测试用例名称** | | 模板渲染节点输出后不清空字段功能正确性测试 | **测试用例标识** | | TC049 | | **测试需求标识** | | TR2.12 | |
| **简要描述** | | 测试模板渲染节点能否根据配置，在输出时保留已填字段的功能，以便只更新模板中的一个字段 | | | | | | | | |
| **前提和约束** | | 无 | | | | | | | | |
| **测试方法** | | 单元测试 | | | | | | | | |
| **测试过程描述** | | | | | | | | | | |
| **序号** | **测试步骤** | | **输入** | **测试结果** | | | | **评价准则** | | **测试结论** |
| **预期结果** | | **实际结果** | |
| 1 | 建立node-red测试环境 | |  |  | |  | |  | |  |
| 2 | 创建含有一个模板渲染节点和一个helper节点的工作流，配置leftDelimiter:"{{",rightDelimiter:"}}",clearWhenOutput:"false",outputWhenAllSet:"true",template:"#Header\n##Header2\nToday is {{date}}\nThis is image ![]({{image}})", | |  |  | |  | |  | |  |
| 3 | 部署工作流 | |  |  | |  | |  | |  |
| 4 | 设置helper节点的消息处理回调函数用于后续判断 | |  |  | |  | |  | |  |
| 5 | 向模板渲染节点输入数据 | | {payload:{date:"May 13"}} |  | |  | |  | |  |
| 6 | 向模板渲染节点输入数据 | | {payload:{image:"https://test.com/img"}} | helper收到消息，payload为"#Header\n##Header2\nToday is May 13\nThis is image ![](https://test.com/img)" | | helper收到消息，payload为"#Header\n##Header2\nToday is May 13\nThis is image ![](https://test.com/img)" | | 本次输入有输出，且数据一致 | | 通过 |
| 7 | 向模板渲染节点输入数据 | | {payload:{image:"https://test2.com/img"}} | helper收到消息，payload为"#Header\n##Header2\nToday is May 13\nThis is image ![](https://test2.com/img)" | | helper收到消息，payload为"#Header\n##Header2\nToday is May 13\nThis is image ![](https://test2.com/img)" | | 本次输入有输出，且数据一致 | | 通过 |
| 8 | 停止node-red测试环境 | |  |  | |  | |  | |  |
| **备注** | | 无 | | | | | | | | |

* + 1. 模板渲染节点解析模板时忽略无效占位符功能正确性测试

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **测试用例名称** | | 模板渲染节点解析模板时忽略无效占位符功能正确性测试 | **测试用例标识** | | TC050 | | **测试需求标识** | | TR2.12 | |
| **简要描述** | | 测试模板渲染节点能否在解析模板的时候忽略无效占位符，例如若干空格，这部分字符串将不被视为需要替换的字段 | | | | | | | | |
| **前提和约束** | | 无 | | | | | | | | |
| **测试方法** | | 单元测试 | | | | | | | | |
| **测试过程描述** | | | | | | | | | | |
| **序号** | **测试步骤** | | **输入** | **测试结果** | | | | **评价准则** | | **测试结论** |
| **预期结果** | | **实际结果** | |
| 1 | 建立node-red测试环境 | |  |  | |  | |  | |  |
| 2 | 创建含有一个模板渲染节点和一个helper节点的工作流，配置leftDelimiter:"{{",rightDelimiter:"}}",clearWhenOutput:"true",outputWhenAllSet:"true",template:"#Header\n##Header2\nToday is {{date}}\nThis is image ![]({{ }})", | |  |  | |  | |  | |  |
| 3 | 部署工作流 | |  |  | |  | |  | |  |
| 4 | 设置helper节点的消息处理回调函数用于后续判断 | |  |  | |  | |  | |  |
| 5 | 向模板渲染节点输入数据 | | {payload:{date:"May 13"}} | helper收到消息，payload为"#Header\n##Header2\nToday is May 13\nThis is image ![]({{ }})" | | helper收到消息，payload为"#Header\n##Header2\nToday is May 13\nThis is image ![]({{ }})" | | 数据一致 | | 通过 |
| 6 | 停止node-red测试环境 | |  |  | |  | |  | |  |
| **备注** | | 无 | | | | | | | | |

* + 1. 模板渲染节点不等待所有字段功能正确性测试

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **测试用例名称** | | 模板渲染节点不等待所有字段功能正确性测试 | **测试用例标识** | | TC051 | | **测试需求标识** | | TR2.12 | |
| **简要描述** | | 测试模板渲染节点能否根据配置，当有输入时不等待所有字段都被设置便输出部分替换的结果 | | | | | | | | |
| **前提和约束** | | 无 | | | | | | | | |
| **测试方法** | | 单元测试 | | | | | | | | |
| **测试过程描述** | | | | | | | | | | |
| **序号** | **测试步骤** | | **输入** | **测试结果** | | | | **评价准则** | | **测试结论** |
| **预期结果** | | **实际结果** | |
| 1 | 建立node-red测试环境 | |  |  | |  | |  | |  |
| 2 | 创建含有一个模板渲染节点和一个helper节点的工作流，配置leftDelimiter:"{{",rightDelimiter:"}}",clearWhenOutput:"false",outputWhenAllSet:"false",template:"#Header\n##Header2\nToday is {{date}}\nThis is image ![]({{image}})", | |  |  | |  | |  | |  |
| 3 | 部署工作流 | |  |  | |  | |  | |  |
| 4 | 设置helper节点的消息处理回调函数用于后续判断 | |  |  | |  | |  | |  |
| 5 | 向模板渲染节点输入数据 | | {payload:{date:"May 13"}} | helper收到消息，payload为"#Header\n##Header2\nToday is May 13\nThis is image ![]({{image}})" | | helper收到消息，payload为"#Header\n##Header2\nToday is May 13\nThis is image ![]({{image}})" | | 数据一致 | | 通过 |
| 6 | 向模板渲染节点输入数据 | | {payload:{image:"https://test.com/img"}} | helper收到消息，payload为"#Header\n##Header2\nToday is May 13\nThis is image ![](https://test.com/img)" | | helper收到消息，payload为"#Header\n##Header2\nToday is May 13\nThis is image ![](https://test.com/img)" | | 数据一致 | | 通过 |
| 7 | 停止node-red测试环境 | |  |  | |  | |  | |  |
| **备注** | | 无 | | | | | | | | |

## 创建窗口节点测试

* + 1. 窗口节点部署测试

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **测试用例名称** | | 窗口节点部署测试 | **测试用例标识** | | TC052 | | **测试需求标识** | | TR2.13 | |
| **简要描述** | | 测试窗口节点能否部署 | | | | | | | | |
| **前提和约束** | | 无 | | | | | | | | |
| **测试方法** | | 冒烟测试 | | | | | | | | |
| **测试过程描述** | | | | | | | | | | |
| **序号** | **测试步骤** | | **输入** | **测试结果** | | | | **评价准则** | | **测试结论** |
| **预期结果** | | **实际结果** | |
| 1 | 建立node-red测试环境 | |  |  | |  | |  | |  |
| 2 | 创建含有一个name属性为"window"，其他属性不配置的窗口节点的工作流 | |  |  | |  | |  | |  |
| 3 | 部署工作流 | |  |  | |  | |  | |  |
| 4 | 获取窗口节点的name属性 | |  | 属性值为"window" | | 属性值为"window" | | 属性值一致 | | 通过 |
| 5 | 停止node-red测试环境 | |  |  | |  | |  | |  |
| **备注** | | 无 | | | | | | | | |

* + 1. 窗口节点配置异常处理测试

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **测试用例名称** | | 窗口节点配置异常处理测试 | **测试用例标识** | | TC053 | | **测试需求标识** | | TR2.13 | |
| **简要描述** | | 测试窗口节点能否在配置错误时报错 | | | | | | | | |
| **前提和约束** | | 无 | | | | | | | | |
| **测试方法** | | 单元测试 | | | | | | | | |
| **测试过程描述** | | | | | | | | | | |
| **序号** | **测试步骤** | | **输入** | **测试结果** | | | | **评价准则** | | **测试结论** |
| **预期结果** | | **实际结果** | |
| 1 | 建立node-red测试环境 | |  |  | |  | |  | |  |
| 2 | 创建含有一个无配置的窗口节点的工作流，name属性为window | |  |  | |  | |  | |  |
| 3 | 部署工作流 | |  | 节点报错，错误信息开头为"unknown mode" | | 节点报错，错误信息开头为"unknown mode" | | 报错且错误信息一致 | | 通过 |
| 4 | 停止node-red测试环境 | |  |  | |  | |  | |  |
| **备注** | | 无 | | | | | | | | |

* + 1. 窗口节点session模式功能正确性测试

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **测试用例名称** | | 窗口节点session模式功能正确性测试 | **测试用例标识** | | TC054 | | **测试需求标识** | | TR2.13 | |
| **简要描述** | | 测试session窗口节点能否按配置的最小时间间隔划分时序输入的数据 | | | | | | | | |
| **前提和约束** | | 无 | | | | | | | | |
| **测试方法** | | 单元测试 | | | | | | | | |
| **测试过程描述** | | | | | | | | | | |
| **序号** | **测试步骤** | | **输入** | **测试结果** | | | | **评价准则** | | **测试结论** |
| **预期结果** | | **实际结果** | |
| 1 | 建立node-red测试环境 | |  |  | |  | |  | |  |
| 2 | 创建含有一个session窗口节点和一个helper节点的工作流，session的时间间隔参数为1000ms | |  |  | |  | |  | |  |
| 3 | 部署工作流，标记为t=0ms时间 | |  |  | |  | |  | |  |
| 4 | 设置helper节点的消息处理回调函数用于后续判断 | |  |  | |  | |  | |  |
| 5 | 在t=0ms时，向session窗口节点输入数据 | | {payload:0} |  | |  | |  | |  |
| 6 | 在t=500ms时，向session窗口节点输入数据 | |  |  | |  | |  | |  |
| 7 | 在t=1500ms时 | |  | helper收到数据{payload:[0,0]} | | helper收到数据{payload:[0,0]} | | 时间点正确，数据一致 | | 通过 |
| 8 | 在t=1600ms时，向session窗口节点输入数据 | | {payload:0} |  | |  | |  | |  |
| 9 | 在t=2000ms时，向session窗口节点输入数据 | | {payload:0} |  | |  | |  | |  |
| 10 | 在t=2900ms时，向session窗口节点输入数据 | | {payload:0} |  | |  | |  | |  |
| 11 | 在t=3900ms时 | |  | helper收到数据{payload:[0,0,0]} | | helper收到数据{payload:[0,0,0]} | | 时间点正确，数据一致 | | 通过 |
| 12 | 在t=4000ms时，向session窗口节点输入数据 | | {payload:0} |  | |  | |  | |  |
| 13 | 在t=5000ms时 | |  | helper收到数据{payload:[0]} | | helper收到数据{payload:[0]} | | 时间点正确，数据一致 | | 通过 |
| 14 | 停止node-red测试环境 | |  |  | |  | |  | |  |
| **备注** | | 由于js中的时间函数不是精确的，对输出中的时间戳不做精确要求，在表格中忽略 | | | | | | | | |

* + 1. 窗口节点session模式首次输入激活测试

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **测试用例名称** | | 窗口节点session模式首次输入激活测试 | **测试用例标识** | | TC055 | | **测试需求标识** | | TR2.13 | |
| **简要描述** | | 测试窗口节点session模式应当由首次输入激活，在未激活时，不应该输出任何数据 | | | | | | | | |
| **前提和约束** | | 无 | | | | | | | | |
| **测试方法** | | 单元测试 | | | | | | | | |
| **测试过程描述** | | | | | | | | | | |
| **序号** | **测试步骤** | | **输入** | **测试结果** | | | | **评价准则** | | **测试结论** |
| **预期结果** | | **实际结果** | |
| 1 | 建立node-red测试环境 | |  |  | |  | |  | |  |
| 2 | 创建含有一个session窗口节点和一个helper节点的工作流，session的时间间隔参数为1000ms | |  |  | |  | |  | |  |
| 3 | 部署工作流，标记为t=0ms时间 | |  |  | |  | |  | |  |
| 4 | 设置helper节点的消息处理回调函数用于后续判断，标记为t=0ms时间 | |  |  | |  | |  | |  |
| 5 | 在t=2000ms时，向session窗口节点输入数据 | | {payload:0} |  | |  | |  | |  |
| 6 | 在t=2500ms时，向session窗口节点输入数据 | | {payload:0} |  | |  | |  | |  |
| 7 | 在t=3500ms时 | |  | helper收到数据{payload:[0,0]} | | helper收到数据{payload:[0,0]} | | 时间点正确，数据一致 | | 通过 |
| 8 | 在t=3600ms时，向session窗口节点输入数据 | | {payload:0} |  | |  | |  | |  |
| 9 | 在t=4600ms时 | |  | helper收到数据{payload:[0]} | | helper收到数据{payload:[0]} | | 时间点正确，数据一致 | | 通过 |
| 10 | 停止node-red测试环境 | |  |  | |  | |  | |  |
| **备注** | | 由于js中的时间函数不是精确的，对输出中的时间戳不做精确要求，在表格中忽略 | | | | | | | | |

* + 1. 窗口节点session模式配置异常处理测试

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **测试用例名称** | | 窗口节点session模式配置异常处理测试 | **测试用例标识** | | TC056 | | **测试需求标识** | | TR2.13 | |
| **简要描述** | | 测试窗口节点session模式能否在配置不当时进行报错 | | | | | | | | |
| **前提和约束** | | 无 | | | | | | | | |
| **测试方法** | | 单元测试 | | | | | | | | |
| **测试过程描述** | | | | | | | | | | |
| **序号** | **测试步骤** | | **输入** | **测试结果** | | | | **评价准则** | | **测试结论** |
| **预期结果** | | **实际结果** | |
| 1 | 建立node-red测试环境 | |  |  | |  | |  | |  |
| 2 | 创建含有一个session窗口节点的工作流，session的时间间隔参数为0ms | |  |  | |  | |  | |  |
| 3 | 部署工作流 | |  | 节点报错，错误信息开头为"invalid window size" | | 节点报错，错误信息开头为"invalid window size" | | 报错且错误信息一致 | | 通过 |
| 4 | 停止node-red测试环境 | |  |  | |  | |  | |  |
| **备注** | | 由于js中的时间函数不是精确的，对输出中的时间戳不做精确要求，在表格中忽略 | | | | | | | | |

* + 1. 窗口节点tumbling时间模式功能正确性测试

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **测试用例名称** | | 窗口节点tumbling时间模式功能正确性测试 | **测试用例标识** | | TC057 | | **测试需求标识** | | TR2.13 | |
| **简要描述** | | 测试窗口节点tumbling时间模式能否按配置的固定时长划分时序输入 | | | | | | | | |
| **前提和约束** | | 无 | | | | | | | | |
| **测试方法** | | 单元测试 | | | | | | | | |
| **测试过程描述** | | | | | | | | | | |
| **序号** | **测试步骤** | | **输入** | **测试结果** | | | | **评价准则** | | **测试结论** |
| **预期结果** | | **实际结果** | |
| 1 | 建立node-red测试环境 | |  |  | |  | |  | |  |
| 2 | 创建含有一个tumbling时间窗口节点和一个helper节点的工作流，窗口时长参数为1000ms | |  |  | |  | |  | |  |
| 3 | 部署工作流，标记为t=0ms时间 | |  |  | |  | |  | |  |
| 4 | 设置helper节点的消息处理回调函数用于后续判断 | |  |  | |  | |  | |  |
| 5 | 在t=0ms时，向tumbling时间窗口节点输入数据 | | {payload:0} |  | |  | |  | |  |
| 6 | 在t=500ms时，向tumbling时间窗口节点输入数据 | | {payload:0} |  | |  | |  | |  |
| 7 | 在t=1000ms时 | |  | helper收到数据{payload:[0,0]} | | helper收到数据{payload:[0,0]} | | 时间点正确，数据一致 | | 通过 |
| 8 | 在t=1100ms时，向tumbling时间窗口节点输入数据 | | {payload:0} |  | |  | |  | |  |
| 9 | 在t=1800ms时，向tumbling时间窗口节点输入数据 | | {payload:0} |  | |  | |  | |  |
| 10 | 在t=1900ms时，向tumbling时间窗口节点输入数据 | | {payload:0} |  | |  | |  | |  |
| 11 | 在t=2000ms时 | |  | helper收到数据{payload:[0,0,0]} | | helper收到数据{payload:[0,0,0]} | | 时间点正确，数据一致 | | 通过 |
| 12 | 在t=2500ms时，向tumbling时间窗口节点输入数据 | | {payload:0} |  | |  | |  | |  |
| 13 | 在t=3000ms时 | |  | helper收到数据{payload:[0]} | | helper收到数据{payload:[0]} | | 时间点正确，数据一致 | | 通过 |
| 14 | 停止node-red测试环境 | |  |  | |  | |  | |  |
| **备注** | | 由于js中的时间函数不是精确的，对输出中的时间戳不做精确要求，在表格中忽略 | | | | | | | | |

* + 1. 窗口节点tumbling时间模式无数据测试

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **测试用例名称** | | 窗口节点tumbling时间模式无数据测试 | **测试用例标识** | | TC058 | | **测试需求标识** | | TR2.13 | |
| **简要描述** | | 测试窗口节点tumbling时间模式在无数据的时间窗口内是否有输出，功能正确是应无输出，因为该模式下窗口数据的统计从部署便开始，对于没有任何数据到来的窗口应该不输出数据 | | | | | | | | |
| **前提和约束** | | 无 | | | | | | | | |
| **测试方法** | | 单元测试 | | | | | | | | |
| **测试过程描述** | | | | | | | | | | |
| **序号** | **测试步骤** | | **输入** | **测试结果** | | | | **评价准则** | | **测试结论** |
| **预期结果** | | **实际结果** | |
| 1 | 建立node-red测试环境 | |  |  | |  | |  | |  |
| 2 | 创建含有一个tumbling时间窗口节点和一个helper节点的工作流，窗口时长参数为1000ms | |  |  | |  | |  | |  |
| 3 | 部署工作流，标记为t=0ms时间 | |  |  | |  | |  | |  |
| 4 | 设置helper节点的消息处理回调函数用于后续判断 | |  |  | |  | |  | |  |
| 5 | 在t=1100ms时，向tumbling时间窗口节点输入数据 | | {payload:0} |  | |  | |  | |  |
| 6 | 在t=1200ms时，向tumbling时间窗口节点输入数据 | | {payload:0} |  | |  | |  | |  |
| 7 | 在t=2000ms时 | |  | helper收到数据{payload:[0,0]} | | helper收到数据{payload:[0,0]} | | 时间点正确，数据一致 | | 通过 |
| 8 | 停止node-red测试环境 | |  |  | |  | |  | |  |
| **备注** | | 由于js中的时间函数不是精确的，对输出中的时间戳不做精确要求，在表格中忽略 | | | | | | | | |

* + 1. 窗口节点tumbling时间模式配置异常处理测试

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **测试用例名称** | | 窗口节点tumbling时间模式配置异常处理测试 | **测试用例标识** | | TC059 | | **测试需求标识** | | TR2.13 | |
| **简要描述** | | 测试窗口节点tumbling时间模式能否在配置不当时进行报错 | | | | | | | | |
| **前提和约束** | | 无 | | | | | | | | |
| **测试方法** | | 单元测试 | | | | | | | | |
| **测试过程描述** | | | | | | | | | | |
| **序号** | **测试步骤** | | **输入** | **测试结果** | | | | **评价准则** | | **测试结论** |
| **预期结果** | | **实际结果** | |
| 1 | 建立node-red测试环境 | |  |  | |  | |  | |  |
| 2 | 创建含有一个tumbling时间窗口节点的工作流，窗口时间长度参数为-3600ms | |  |  | |  | |  | |  |
| 3 | 部署工作流 | |  | 节点报错，错误信息开头为"invalid window size" | | 节点报错，错误信息开头为"invalid window size" | | 报错且错误信息一致 | | 通过 |
| 4 | 停止node-red测试环境 | |  |  | |  | |  | |  |
| **备注** | | 由于js中的时间函数不是精确的，对输出中的时间戳不做精确要求，在表格中忽略 | | | | | | | | |

* + 1. 窗口节点tumbling数据模式功能正确性测试

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **测试用例名称** | | 时间窗口节点tumbling数据模式功能正确性测试 | **测试用例标识** | | TC060 | | **测试需求标识** | | TR2.13 | |
| **简要描述** | | 测试窗口节点tumbling数据模式能否按配置的固定时长划分时序输入 | | | | | | | | |
| **前提和约束** | | 无 | | | | | | | | |
| **测试方法** | | 单元测试 | | | | | | | | |
| **测试过程描述** | | | | | | | | | | |
| **序号** | **测试步骤** | | **输入** | **测试结果** | | | | **评价准则** | | **测试结论** |
| **预期结果** | | **实际结果** | |
| 1 | 建立node-red测试环境 | |  |  | |  | |  | |  |
| 2 | 创建含有一个tumbling数量窗口节点和一个helper节点的工作流，窗口数量参数为2 | |  |  | |  | |  | |  |
| 3 | 部署工作流 | |  |  | |  | |  | |  |
| 4 | 设置helper节点的消息处理回调函数用于后续判断 | |  |  | |  | |  | |  |
| 5 | 在t=0ms时，向tumbling时间窗口节点输入数据 | | {payload:0} |  | |  | |  | |  |
| 6 | 在t=500ms时，向tumbling时间窗口节点输入数据 | | {payload:0} | helper收到数据{payload:[0,0]} | | helper收到数据{payload:[0,0]} | | 输入后立刻收到消息，数据一致 | | 通过 |
| 7 | 在t=1100ms时，向tumbling时间窗口节点输入数据 | | {payload:0} |  | |  | |  | |  |
| 8 | 在t=1800ms时，向tumbling时间窗口节点输入数据 | | {payload:0} | helper收到数据{payload:[0,0]} | | helper收到数据{payload:[0,0]} | | 输入后立刻收到消息，数据一致 | | 通过 |
| 9 | 在t=1900ms时，向tumbling时间窗口节点输入数据 | | {payload:0} |  | |  | |  | |  |
| 10 | 在t=2500ms时，向tumbling时间窗口节点输入数据 | | {payload:0} | helper收到数据{payload:[0,0]} | | helper收到数据{payload:[0,0]} | | 输入后立刻收到消息，数据一致 | | 通过 |
| 11 | 停止node-red测试环境 | |  |  | |  | |  | |  |
| **备注** | | 由于js中的时间函数不是精确的，对输出中的时间戳不做精确要求，在表格中忽略 | | | | | | | | |

* + 1. 窗口节点tumbling数量模式配置异常处理测试

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **测试用例名称** | | 窗口节点tumbling数量模式配置异常处理测试 | **测试用例标识** | | TC061 | | **测试需求标识** | | TR2.13 | |
| **简要描述** | | 测试窗口节点tumbling数量模式能否在配置不当时进行报错 | | | | | | | | |
| **前提和约束** | | 无 | | | | | | | | |
| **测试方法** | | 单元测试 | | | | | | | | |
| **测试过程描述** | | | | | | | | | | |
| **序号** | **测试步骤** | | **输入** | **测试结果** | | | | **评价准则** | | **测试结论** |
| **预期结果** | | **实际结果** | |
| 1 | 建立node-red测试环境 | |  |  | |  | |  | |  |
| 2 | 创建含有一个tumbling数量窗口节点的工作流，窗口数量长度参数为-100 | |  |  | |  | |  | |  |
| 3 | 部署工作流 | |  | 节点报错，错误信息开头为"invalid window size" | | 节点报错，错误信息开头为"invalid window size" | | 报错且错误信息一致 | | 通过 |
| 4 | 停止node-red测试环境 | |  |  | |  | |  | |  |
| **备注** | | 由于js中的时间函数不是精确的，对输出中的时间戳不做精确要求，在表格中忽略 | | | | | | | | |

* + 1. 窗口节点sliding时间模式功能正确性测试

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **测试用例名称** | | 窗口节点sliding时间模式功能正确性测试 | **测试用例标识** | | TC062 | | **测试需求标识** | | TR2.13 | |
| **简要描述** | | 测试窗口节点sliding时间模式能否按配置的窗口时长和滑动时长生成时序输入的正确窗口输出 | | | | | | | | |
| **前提和约束** | | 无 | | | | | | | | |
| **测试方法** | | 单元测试 | | | | | | | | |
| **测试过程描述** | | | | | | | | | | |
| **序号** | **测试步骤** | | **输入** | **测试结果** | | | | **评价准则** | | **测试结论** |
| **预期结果** | | **实际结果** | |
| 1 | 建立node-red测试环境 | |  |  | |  | |  | |  |
| 2 | 创建含有一个sliding时间窗口节点和一个helper节点的工作流，时间窗长参数为3000ms，滑动时长为1000ms | |  |  | |  | |  | |  |
| 3 | 部署工作流，标记为t=0ms时刻 | |  |  | |  | |  | |  |
| 4 | 设置helper节点的消息处理回调函数用于后续判断 | |  |  | |  | |  | |  |
| 5 | 在t=500ms时，向tumbling时间窗口节点输入数据 | | {payload:0} |  | |  | |  | |  |
| 6 | 在t=1500ms时，向tumbling时间窗口节点输入数据 | | {payload:0} |  | |  | |  | |  |
| 7 | 在t=2500ms时，向tumbling时间窗口节点输入数据 | | {payload:0} |  | |  | |  | |  |
| 8 | 在t=3000ms时 | |  | helper收到数据{payload:[0,0,0]} | | helper收到数据{payload:[0,0,0]} | | 时间点正确，数据一致 | | 通过 |
| 9 | 在t=3500ms时，向tumbling时间窗口节点输入数据 | | {payload:0} |  | |  | |  | |  |
| 10 | 在t=4000ms时 | |  | helper收到数据{payload:[0,0,0]} | | helper收到数据{payload:[0,0,0]} | | 时间点正确，数据一致 | | 通过 |
| 11 | 在t=4500ms时，向tumbling时间窗口节点输入数据 | | {payload:0} |  | |  | |  | |  |
| 12 | 在t=5000ms时 | |  | helper收到数据{payload:[0,0,0]} | | helper收到数据{payload:[0,0,0]} | | 时间点正确，数据一致 | | 通过 |
| 13 | 在t=6000ms时 | |  | helper收到数据{payload:[0,0]} | | helper收到数据{payload:[0,0]} | | 时间点正确，数据一致 | | 通过 |
| 14 | 在t=7000ms时 | |  | helper收到数据{payload:[0]} | | helper收到数据{payload:[0]} | | 时间点正确，数据一致 | | 通过 |
| 15 | 停止node-red测试环境 | |  |  | |  | |  | |  |
| **备注** | | 由于js中的时间函数不是精确的，对输出中的时间戳不做精确要求，在表格中忽略 | | | | | | | | |

* + 1. 窗口节点sliding时间模式窗口时长配置异常处理测试

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **测试用例名称** | | 窗口节点sliding时间模式窗口时长配置异常处理测试 | **测试用例标识** | | TC063 | | **测试需求标识** | | TR2.13 | |
| **简要描述** | | 测试窗口节点sliding时间模式能否在窗口时长配置不当时进行报错 | | | | | | | | |
| **前提和约束** | | 无 | | | | | | | | |
| **测试方法** | | 单元测试 | | | | | | | | |
| **测试过程描述** | | | | | | | | | | |
| **序号** | **测试步骤** | | **输入** | **测试结果** | | | | **评价准则** | | **测试结论** |
| **预期结果** | | **实际结果** | |
| 1 | 建立node-red测试环境 | |  |  | |  | |  | |  |
| 2 | 创建含有一个sliding时间窗口节点的工作流，窗口时长参数为"asdfgh"，滑动时长为1000ms | |  |  | |  | |  | |  |
| 3 | 部署工作流 | |  | 节点报错，错误信息开头为"invalid window size" | | 节点报错，错误信息开头为"invalid window size" | | 报错且错误信息一致 | | 通过 |
| 4 | 停止node-red测试环境 | |  |  | |  | |  | |  |
| **备注** | |  | | | | | | | | |

* + 1. 窗口节点sliding时间模式滑动时长配置异常处理测试

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **测试用例名称** | | 窗口节点sliding时间模式滑动时长配置异常处理测试 | **测试用例标识** | | TC064 | | **测试需求标识** | | TR2.13 | |
| **简要描述** | | 测试窗口节点sliding时间模式能否在滑动时长配置不当时进行报错 | | | | | | | | |
| **前提和约束** | | 无 | | | | | | | | |
| **测试方法** | | 单元测试 | | | | | | | | |
| **测试过程描述** | | | | | | | | | | |
| **序号** | **测试步骤** | | **输入** | **测试结果** | | | | **评价准则** | | **测试结论** |
| **预期结果** | | **实际结果** | |
| 1 | 建立node-red测试环境 | |  |  | |  | |  | |  |
| 2 | 创建含有一个sliding时间窗口节点的工作流，窗口时长参数为3000ms，滑动时长为undefined | |  |  | |  | |  | |  |
| 3 | 部署工作流 | |  | 节点报错，错误信息开头为"invalid slide size" | | 节点报错，错误信息开头为"invalid slide size" | | 报错且错误信息一致 | | 通过 |
| 4 | 停止node-red测试环境 | |  |  | |  | |  | |  |
| **备注** | |  | | | | | | | | |

* + 1. 窗口节点sliding数量模式功能正确性测试

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **测试用例名称** | | 窗口节点sliding数量模式功能正确性测试 | **测试用例标识** | | TC065 | | **测试需求标识** | | TR2.13 | |
| **简要描述** | | 测试窗口节点sliding数量模式能否按配置的窗口时长和滑动时长生成时序输入的正确窗口输出 | | | | | | | | |
| **前提和约束** | | 无 | | | | | | | | |
| **测试方法** | | 单元测试 | | | | | | | | |
| **测试过程描述** | | | | | | | | | | |
| **序号** | **测试步骤** | | **输入** | **测试结果** | | | | **评价准则** | | **测试结论** |
| **预期结果** | | **实际结果** | |
| 1 | 建立node-red测试环境 | |  |  | |  | |  | |  |
| 2 | 创建含有一个sliding数量窗口节点和一个helper节点的工作流，窗口数量参数为3，滑动数量为1 | |  |  | |  | |  | |  |
| 3 | 部署工作流 | |  |  | |  | |  | |  |
| 4 | 设置helper节点的消息处理回调函数用于后续判断 | |  |  | |  | |  | |  |
| 5 | 向tumbling时间窗口节点输入数据 | | {payload:0} |  | |  | |  | |  |
| 6 | 向tumbling时间窗口节点输入数据 | | {payload:0} |  | |  | |  | |  |
| 7 | 向tumbling时间窗口节点输入数据 | | {payload:0} | helper收到数据{payload:[0,0,0]} | | helper收到数据{payload:[0,0,0]} | | 输入后立刻收到输出，数据一致 | | 通过 |
| 8 | 向tumbling时间窗口节点输入数据 | | {payload:0} | helper收到数据{payload:[0,0,0]} | | helper收到数据{payload:[0,0,0]} | | 输入后立刻收到输出，数据一致 | | 通过 |
| 9 | 向tumbling时间窗口节点输入数据 | | {payload:0} | helper收到数据{payload:[0,0,0]} | | helper收到数据{payload:[0,0,0]} | | 输入后立刻收到输出，数据一致 | | 通过 |
| 10 | 停止node-red测试环境 | |  |  | |  | |  | |  |
| **备注** | | 由于js中的时间函数不是精确的，对输出中的时间戳不做精确要求，在表格中忽略 | | | | | | | | |

* + 1. 窗口节点sliding数量模式窗口数量配置异常处理测试

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **测试用例名称** | | 窗口节点sliding数量模式窗口数量配置异常处理测试 | **测试用例标识** | | TC066 | | **测试需求标识** | | TR2.13 | |
| **简要描述** | | 测试窗口节点sliding数量模式能否在窗口数量配置不当时进行报错 | | | | | | | | |
| **前提和约束** | | 无 | | | | | | | | |
| **测试方法** | | 单元测试 | | | | | | | | |
| **测试过程描述** | | | | | | | | | | |
| **序号** | **测试步骤** | | **输入** | **测试结果** | | | | **评价准则** | | **测试结论** |
| **预期结果** | | **实际结果** | |
| 1 | 建立node-red测试环境 | |  |  | |  | |  | |  |
| 2 | 创建含有一个sliding时间窗口节点的工作流，窗口时长参数为"qwer"，滑动时长为1 | |  |  | |  | |  | |  |
| 3 | 部署工作流 | |  | 节点报错，错误信息开头为"invalid window size" | | 节点报错，错误信息开头为"invalid window size" | | 报错且错误信息一致 | | 通过 |
| 4 | 停止node-red测试环境 | |  |  | |  | |  | |  |
| **备注** | |  | | | | | | | | |

* + 1. 窗口节点sliding数量模式滑动数量配置异常处理测试

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **测试用例名称** | | 窗口节点sliding数量模式滑动数量配置异常处理测试 | **测试用例标识** | | TC067 | | **测试需求标识** | | TR2.13 | |
| **简要描述** | | 测试窗口节点sliding数量模式能否在滑动数量配置不当时进行报错 | | | | | | | | |
| **前提和约束** | | 无 | | | | | | | | |
| **测试方法** | | 单元测试 | | | | | | | | |
| **测试过程描述** | | | | | | | | | | |
| **序号** | **测试步骤** | | **输入** | **测试结果** | | | | **评价准则** | | **测试结论** |
| **预期结果** | | **实际结果** | |
| 1 | 建立node-red测试环境 | |  |  | |  | |  | |  |
| 2 | 创建含有一个sliding时间窗口节点的工作流，窗口数量参数为3，滑动数量为null | |  |  | |  | |  | |  |
| 3 | 部署工作流 | |  | 节点报错，错误信息开头为"invalid slide size" | | 节点报错，错误信息开头为"invalid slide size" | | 报错且错误信息一致 | | 通过 |
| 4 | 停止node-red测试环境 | |  |  | |  | |  | |  |
| **备注** | |  | | | | | | | | |

# 示例场景测试报告

## 飞书机器人基本功能测试

* + 1. 获取当前CPU使用情况图测试

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **测试用例名称** | | 获取当前CPU使用情况图测试 | **测试用例标识** | | TC068 | | **测试需求标识** | | - | |
| **简要描述** | | 用户能否通过私聊飞书机器人获取到当前node-red服务器CPU使用情况图 | | | | | | | | |
| **前提和约束** | | 用户拥有飞书账号，且处在飞书机器人的群聊中，能私聊飞书机器人 | | | | | | | | |
| **测试方法** | | 场景测试 | | | | | | | | |
| **测试过程描述** | | | | | | | | | | |
| **序号** | **测试步骤** | | **输入** | **测试结果** | | | | **评价准则** | | **测试结论** |
| **预期结果** | | **实际结果** | |
| 1 | 登录飞书账号 | |  |  | |  | |  | |  |
| 2 | 通过群聊打开飞书机器人私聊界面 | |  |  | |  | |  | |  |
| 3 | 向飞书机器人发送获取CPU图的请求 | | 获取CPU使用情况 | 正在获取CPU使用情况...  当前CPU使用情况图 | | 正在获取CPU使用情况...  当前CPU使用情况图 | | 返回结果 | | 通过 |
| **备注** | | 输入与结果为与飞书机器人私聊的消息 | | | | | | | | |

* + 1. 获取CPU使用情况订阅测试

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **测试用例名称** | | 获取CPU使用情况订阅测试 | **测试用例标识** | | TC069 | | **测试需求标识** | | - | |
| **简要描述** | | 用户能否通过私聊飞书机器人订阅node-red服务器CPU使用情况图 | | | | | | | | |
| **前提和约束** | | 用户拥有飞书账号，且处在飞书机器人的群聊中，能私聊飞书机器人 | | | | | | | | |
| **测试方法** | | 场景测试 | | | | | | | | |
| **测试过程描述** | | | | | | | | | | |
| **序号** | **测试步骤** | | **输入** | **测试结果** | | | | **评价准则** | | **测试结论** |
| **预期结果** | | **实际结果** | |
| 1 | 登录飞书账号 | |  |  | |  | |  | |  |
| 2 | 通过群聊打开飞书机器人私聊界面 | |  |  | |  | |  | |  |
| 3 | 向飞书机器人发送获取CPU使用情况图的请求 | | 开启监控CPU使用情况 | 已提交CPU使用情况订阅申请。将每隔1分钟发送CPU使用情况折线图 | | 已提交CPU使用情况订阅申请。将每隔1分钟发送CPU使用情况折线图 | | 返回提示 | | 通过 |
| 4 | 等待定时的CPU使用情况图发送，确认成功订阅 | |  | 每隔1分钟收到CPU使用情况图 | | 每隔1分钟收到CPU使用情况图 | | 正确周期性收到图片 | | 通过 |
| **备注** | | 输入与结果为与飞书机器人私聊的消息 | | | | | | | | |

* + 1. 关闭CPU使用情况订阅测试

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **测试用例名称** | | 关闭CPU使用情况订阅测试 | **测试用例标识** | | TC070 | | **测试需求标识** | | - | |
| **简要描述** | | 用户能否通过私聊飞书机器人关闭之前订阅的node-red服务器CPU使用情况图 | | | | | | | | |
| **前提和约束** | | 用户拥有飞书账号，且处在飞书机器人的群聊中，能私聊飞书机器人 | | | | | | | | |
| **测试方法** | | 场景测试 | | | | | | | | |
| **测试过程描述** | | | | | | | | | | |
| **序号** | **测试步骤** | | **输入** | **测试结果** | | | | **评价准则** | | **测试结论** |
| **预期结果** | | **实际结果** | |
| 1 | 登录飞书账号 | |  |  | |  | |  | |  |
| 2 | 通过群聊打开飞书机器人私聊界面 | |  |  | |  | |  | |  |
| 3 | 向飞书机器人发送获取CPU使用情况图的请求 | | 开启监控CPU使用情况 | 已提交CPU使用情况订阅申请。将每隔1分钟发送CPU使用情况折线图 | | 已提交CPU使用情况订阅申请。将每隔1分钟发送CPU使用情况折线图 | | 返回提示 | | 通过 |
| 4 | 等待定时的CPU使用情况图发送，确认成功订阅 | |  | 每隔1分钟收到CPU使用情况图 | | 每隔1分钟收到CPU使用情况图 | | 正确周期性收到图片 | | 通过 |
| 5 | 向飞书机器人发送关闭CPU使用情况图的请求 | | 关闭监控CPU使用情况 | 已关闭CPU使用情况订阅 | | 已关闭CPU使用情况订阅 | | 返回提示 | | 通过 |
| 6 | 等待5分钟，确认是否正确关闭订阅 | |  | 不再收到CPU使用情况图 | | 不再收到CPU使用情况图 | | 不再收到订阅 | | 通过 |
| **备注** | | 输入与结果为与飞书机器人私聊的消息 | | | | | | | | |

## 飞书机器人乱序测试

* + 1. 未订阅时申请关闭CPU使用情况订阅测试

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **测试用例名称** | | 未订阅时申请关闭CPU使用情况订阅测试 | **测试用例标识** | | TC071 | | **测试需求标识** | | - | |
| **简要描述** | | 用户在未订阅时私聊飞书机器人发送关闭CPU使用情况订阅请求。 | | | | | | | | |
| **前提和约束** | | 用户拥有飞书账号，且处在飞书机器人的群聊中，能私聊飞书机器人 | | | | | | | | |
| **测试方法** | | 场景测试 | | | | | | | | |
| **测试过程描述** | | | | | | | | | | |
| **序号** | **测试步骤** | | **输入** | **测试结果** | | | | **评价准则** | | **测试结论** |
| **预期结果** | | **实际结果** | |
| 1 | 登录飞书账号 | |  |  | |  | |  | |  |
| 2 | 通过群聊打开飞书机器人私聊界面 | |  |  | |  | |  | |  |
| 3 | 向飞书机器人发送关闭CPU使用情况图的请求 | | 关闭监控CPU使用情况 | 已关闭CPU使用情况订阅 | | 已关闭CPU使用情况订阅 | | 能够容错，正常返回提示 | | 通过 |
| **备注** | | 输入与结果为与飞书机器人私聊的消息 | | | | | | | | |

* + 1. 订阅中获取CPU使用情况测试

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **测试用例名称** | | 订阅中获取CPU使用情况测试 | **测试用例标识** | | TC072 | | **测试需求标识** | | - | |
| **简要描述** | | 用户在订阅CPU使用情况中申请获取当前CPU使用情况图 | | | | | | | | |
| **前提和约束** | | 用户拥有飞书账号，且处在飞书机器人的群聊中，能私聊飞书机器人 | | | | | | | | |
| **测试方法** | | 场景测试 | | | | | | | | |
| **测试过程描述** | | | | | | | | | | |
| **序号** | **测试步骤** | | **输入** | **测试结果** | | | | **评价准则** | | **测试结论** |
| **预期结果** | | **实际结果** | |
| 1 | 登录飞书账号 | |  |  | |  | |  | |  |
| 2 | 通过群聊打开飞书机器人私聊界面 | |  |  | |  | |  | |  |
| 3 | 向飞书机器人发送获取CPU使用情况图的请求 | | 开启监控CPU使用情况 | 已提交CPU使用情况订阅申请。将每隔1分钟发送CPU使用情况折线图 | | 已提交CPU使用情况订阅申请。将每隔1分钟发送CPU使用情况折线图 | | 返回提示 | | 通过 |
| 4 | 等待定时的CPU使用情况图发送，确认成功订阅 | |  | 每隔1分钟收到CPU使用情况图 | | 每隔1分钟收到CPU使用情况图 | | 正确周期性收到图片 | | 通过 |
| 5 | 向飞书机器人发送获取CPU图的请求 | | 获取CPU使用情况 | 正在获取CPU使用情况...  当前CPU使用情况图 | | 正在获取CPU使用情况...  当前CPU使用情况图 | | 返回结果 | | 通过 |
| 6 | 向飞书机器人发送关闭CPU使用情况图的请求 | | 关闭监控CPU使用情况 | 已关闭CPU使用情况订阅 | | 已关闭CPU使用情况订阅 | | 返回提示 | | 通过 |
| 7 | 等待5分钟，确认是否正确关闭订阅 | |  | 不再收到CPU使用情况图 | | 不再收到CPU使用情况图 | | 不再收到订阅 | | 通过 |
| **备注** | | 输入与结果为与飞书机器人私聊的消息 | | | | | | | | |

## 飞书机器人错误输入测试

* + 1. 向飞书机器人发送非预定格式的请求测试

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **测试用例名称** | | 向飞书机器人发送非预定格式的请求测试 | **测试用例标识** | | TC073 | | **测试需求标识** | | - | |
| **简要描述** | | 用户给飞书机器人发送不符合预期的消息 | | | | | | | | |
| **前提和约束** | | 用户拥有飞书账号，且处在飞书机器人的群聊中，能私聊飞书机器人 | | | | | | | | |
| **测试方法** | | 场景测试 | | | | | | | | |
| **测试过程描述** | | | | | | | | | | |
| **序号** | **测试步骤** | | **输入** | **测试结果** | | | | **评价准则** | | **测试结论** |
| **预期结果** | | **实际结果** | |
| 1 | 登录飞书账号 | |  |  | |  | |  | |  |
| 2 | 通过群聊打开飞书机器人私聊界面 | |  |  | |  | |  | |  |
| 3 | 向飞书机器人发送获取CPU图的请求 | | Hello | 消息格式不正确，支持功能如下： 获取CPU使用情况 开启监控CPU使用情况 关闭监控CPU使用情况 | | 消息格式不正确，支持功能如下： 获取CPU使用情况 开启监控CPU使用情况 关闭监控CPU使用情况 | | 返回消息格式不正确提示信息 | | 通过 |
| **备注** | | 输入与结果为与飞书机器人私聊的消息 | | | | | | | | |

# 非功能需求测试报告

## 负载/压力测试

**项目中需要压力测试的节点为飞书相关节点，但由于飞书平台的限制，本项目组无法进行压力测试。**

## 效能测试

与5.1类似，受限于飞书平台，故不进行测试。

## **软件辅助功能测试**

本项目的目标用户不包括障碍用户，故不进行测试。

## **全球化测试**

本项目没有全球化的需求，故不进行测试。

## **兼容性测试**

经验证，本项目开发的拓展节点可以正常在下面的操作系统部署运行：

* Linux
* macOS
* Windows（在Docker中运行）

可以在下面的浏览器中部署运行：

* Chrome
* Internet Explorer 6、7、8、9、10
* Windows Firefox 33
* Mac Firefox 33
* Opera 12.14
* Mac Safari 7

# 综合评价

**在测试环节中，本项目开发的产品通过了准备的所有测试用例，所有拓展节点均可正常部署并正确完成相应功能，且产品在示例场景中表现良好。部分非功能需求测试由于收到飞书平台的限制未能展开，其余表现良好。经验证，软件工程实验I组开发的产品——“对Node-RED工具的消息聚集和图形可视化拓展”已达到交付标准，该产品能够实现需求规格说明书上的功能，满足用户日常使用的需求。**