|  |
| --- |
| 软件工程实验I组 |
| 测试需求规格说明书v1.4.2 |
| 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  2020-5-19 |

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 |
| V1.4.2 | 2020/5/19 | 张雨濛 | 修改非功能需求测试及测试方法部分 | 暴明坤 | 一稿 |
| V1.4.1 | 2020/5/19 | 叶柏威 | 更新图表渲染节点部分测试用例 | 张雨濛 | 一稿 |
| V1.4.0 | 2020/5/19 | 刘子渊 | 根据本组测试需求评审单对测试需求规格说明书进行评审，修改部分问题 | 张雨濛 | 一稿 |
| V1.3.0 | 2020/5/17 | 张雨濛 | 整合文档内容 | 刘子渊 | 一稿 |
| V1.2.0 | 2020/5/17 | 夏欣怡 | 编制场景测试部分测试用例 | 张雨濛 | 一稿 |
| V1.1.0 | 2020/5/16 | 叶柏威 | 编写测试计划内容部分 | 暴明坤 | 一稿 |
| V1.0.0 | 2020/5/16 | 张雨濛 | 整合文档内容，调整格式及排版 | 暴明坤 | 一稿 |
| V0.5.0 | 2020/5/16 | 暴明坤 | 编制消息聚集部分剩余测试用例 | 张雨濛 | 初稿 |
| V0.4.0 | 2020/5/15 | 刘子渊 | 编制消息聚集部分测试用例 | 张雨濛 | 初稿 |
| V0.3.0 | 2020/5/15 | 叶柏威 | 编制可视化部分剩余测试用例 | 张雨濛 | 初稿 |
| V0.2.0 | 2020/5/15 | 胡俊涛 | 编制可视化部分测试用例 | 张雨濛 | 初稿 |
| V0.1.0 | 2020/5/9 | 张雨濛 | 搭建测试需求规格说明书框架 | 暴明坤，夏欣怡，胡俊涛，张雨濛，叶柏威，刘子渊 | 初稿 |

目录

[1 引言 1](#_Toc40777298)

[1.1 目的 1](#_Toc40777299)

[1.2 软件测试的主要内容 1](#_Toc40777300)

[1.3 文档描述 1](#_Toc40777301)

[1.4 测试需求与需求对照表 2](#_Toc40777302)

[2 测试计划 11](#_Toc40777303)

[2.1 测试工具介绍 11](#_Toc40777304)

[2.1.1 单元测试框架 11](#_Toc40777305)

[2.1.2 覆盖率工具 11](#_Toc40777306)

[2.1.3 HTTP服务器的模拟期望库 12](#_Toc40777307)

[2.1.4 Node-Red测试框架 12](#_Toc40777308)

[2.2 测试类型 13](#_Toc40777309)

[2.2.1 白盒测试 13](#_Toc40777310)

[2.2.2 黑盒测试 13](#_Toc40777311)

[2.3 测试目的 13](#_Toc40777312)

[2.4 测试方法 13](#_Toc40777313)

[3 单元测试、功能测试以及集成测试 13](#_Toc40777314)

[3.1 创建飞书激活节点测试 13](#_Toc40777315)

[3.1.1 飞书激活节点测试 13](#_Toc40777316)

[3.2 创建飞书配置节点测试 15](#_Toc40777317)

[3.2.1 飞书配置节点功能测试 15](#_Toc40777318)

[3.3 创建飞书获取群列表节点测试 17](#_Toc40777319)

[3.3.1 飞书获取群列表节点加载测试 17](#_Toc40777320)

[3.3.2 飞书获取群列表节点功能测试 18](#_Toc40777321)

[3.4 创建飞书获得群信息节点测试 21](#_Toc40777322)

[3.4.1 飞书获得群信息节点功能测试 21](#_Toc40777323)

[3.5 创建飞书获得接收消息节点测试 29](#_Toc40777324)

[3.5.1 飞书获得接收消息节点接受私聊文本测试 29](#_Toc40777325)

[3.5.2 飞书获得接收消息节点接受私聊图片测试 32](#_Toc40777326)

[3.5.3 飞书获得接收消息节点接受群聊文本测试 35](#_Toc40777327)

[3.5.4 飞书获得接收消息节点接受群聊图片测试 38](#_Toc40777328)

[3.6 创建飞书获得上传图片节点测试 41](#_Toc40777329)

[3.6.1 飞书获得上传图片节点加载测试 41](#_Toc40777330)

[3.6.2 飞书获得上传图片节点功能测试 43](#_Toc40777331)

[3.7 创建飞书获取图片节点测试 45](#_Toc40777332)

[3.7.1 飞书获取图片节点加载测试 45](#_Toc40777333)

[3.7.2 飞书获取图片节点功能测试 47](#_Toc40777334)

[3.8 创建飞书添加目标节点测试 49](#_Toc40777335)

[3.8.1 飞书添加目标节点部署测试 49](#_Toc40777336)

[3.8.2 飞书添加目标节点正则表达式筛选器测试 50](#_Toc40777337)

[3.8.3 飞书添加目标节点列表筛选器测试 53](#_Toc40777338)

[3.8.4 飞书添加目标节点筛选对象及属性测试 56](#_Toc40777339)

[3.8.5 飞书添加目标节点直接添加模式测试 62](#_Toc40777340)

[3.8.6 飞书添加目标节点初始化测试 65](#_Toc40777341)

[3.8.7 飞书添加目标节点重复添加测试 68](#_Toc40777342)

[3.8.8 飞书添加目标节点异常处理测试 71](#_Toc40777343)

[3.8.9 飞书添加目标节点配置异常处理测试 80](#_Toc40777344)

[3.9 创建飞书移除目标节点测试 86](#_Toc40777345)

[3.9.1 飞书移除目标节点部署测试 86](#_Toc40777346)

[3.9.2 飞书移除目标节点正则表达式筛选器测试 87](#_Toc40777347)

[3.9.3 飞书移除目标节点列表筛选器测试 90](#_Toc40777348)

[3.9.4 飞书移除目标节点筛选对象及属性测试 93](#_Toc40777349)

[3.9.5 飞书移除目标节点初始化测试 99](#_Toc40777350)

[3.9.6 飞书移除目标节点异常处理测试 102](#_Toc40777351)

[3.9.7 飞书移除目标节点配置异常处理测试 106](#_Toc40777352)

[3.10 创建飞书发送消息节点测试 110](#_Toc40777353)

[3.10.1 飞书发送消息节点加载测试 110](#_Toc40777354)

[3.10.2 飞书发送消息节点发送群聊图片测试 111](#_Toc40777355)

[3.10.3 飞书发送消息节点发送群聊文本测试 114](#_Toc40777356)

[3.10.4 飞书发送消息节点发送私聊图片测试 117](#_Toc40777357)

[3.10.5 飞书发送消息节点发送私聊文本测试 120](#_Toc40777358)

[3.11 创建并配置图表渲染节点测试 123](#_Toc40777359)

[3.11.1 图表渲染节点部署测试 123](#_Toc40777360)

[3.11.2 图表渲染节点Line chart绘制测试 124](#_Toc40777361)

[3.11.3 图表渲染节点Bar chart绘制测试 129](#_Toc40777362)

[3.11.4 图表渲染节点Pie chart绘制测试 132](#_Toc40777363)

[3.11.5 图表渲染节点Radar chart绘制测试 134](#_Toc40777364)

[3.11.6 图表渲染节点Polar area chart绘制测试 136](#_Toc40777365)

[3.11.7 图表渲染节点修改图表长宽测试 138](#_Toc40777366)

[3.11.8 图表渲染节点数据窗口功能测试 140](#_Toc40777367)

[3.11.9 图表渲染节点binary输出测试 141](#_Toc40777368)

[3.12 创建并配置模板渲染节点测试 143](#_Toc40777369)

[3.12.1 模板渲染节点部署测试 143](#_Toc40777370)

[3.12.2 模板渲染节点功能正确性测试 144](#_Toc40777371)

[3.12.3 模板渲染节点异常处理测试 146](#_Toc40777372)

[3.12.4 模板渲染节点特殊输入处理测试 148](#_Toc40777373)

[3.12.5 模板渲染节点输出时清空字段功能正确性测试 150](#_Toc40777374)

[3.12.6 模板渲染节点输出后不清空字段功能正确性测试 152](#_Toc40777375)

[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 155](#_Toc40777376)

[3.12.8 模板渲染节点不等待所有字段功能正确性测试 156](#_Toc40777377)

[3.13 创建并配置窗口节点测试 159](#_Toc40777378)

[3.13.1 窗口节点部署测试 159](#_Toc40777379)

[3.13.2 窗口节点配置异常处理测试 160](#_Toc40777380)

[3.13.3 窗口节点session模式功能正确性测试 161](#_Toc40777381)

[3.13.4 窗口节点session模式首次输入激活测试 163](#_Toc40777382)

[3.13.5 窗口节点session模式配置异常处理测试 165](#_Toc40777383)

[3.13.6 窗口节点tumbling时间模式功能正确性测试 167](#_Toc40777384)

[3.13.7 窗口节点tumbling时间模式无数据测试 169](#_Toc40777385)

[3.13.8 窗口节点tumbling时间模式配置异常处理测试 171](#_Toc40777386)

[3.13.9 窗口节点tumbling数据模式功能正确性测试 172](#_Toc40777387)

[3.13.10 窗口节点tumbling数量模式配置异常处理测试 175](#_Toc40777388)

[3.13.11 窗口节点sliding时间模式功能正确性测试 176](#_Toc40777389)

[3.13.12 窗口节点sliding时间模式窗口时长配置异常处理测试 179](#_Toc40777390)

[3.13.13 窗口节点sliding时间模式滑动时长配置异常处理测试 181](#_Toc40777391)

[3.13.14 窗口节点sliding数量模式功能正确性测试 182](#_Toc40777392)

[3.13.15 窗口节点sliding数量模式窗口数量配置异常处理测试 184](#_Toc40777393)

[3.13.16 窗口节点sliding数量模式滑动数量配置异常处理测试 185](#_Toc40777394)

[3.14 创建获取CPU使用率节点测试 187](#_Toc40777395)

[3.14.1 获取CPU使用率测试 187](#_Toc40777396)

[4 场景测试 189](#_Toc40777397)

[4.1 飞书机器人基本功能测试 189](#_Toc40777398)

[4.1.1 获取当前CPU使用情况图测试 189](#_Toc40777399)

[4.1.2 获取CPU使用情况订阅测试 190](#_Toc40777400)

[4.1.3 关闭CPU使用情况订阅测试 191](#_Toc40777401)

[4.2 飞书机器人乱序测试 193](#_Toc40777402)

[4.2.1 未订阅时申请关闭CPU使用情况订阅测试 193](#_Toc40777403)

[4.2.2 订阅中获取CPU使用情况测试 194](#_Toc40777404)

[4.3 飞书机器人错误输入测试 196](#_Toc40777405)

[4.3.1 向飞书机器人发送非预定格式的请求测试 196](#_Toc40777406)

[5 系统测试 197](#_Toc40777407)

[6 AB测试 198](#_Toc40777408)

[7 非功能需求测试 198](#_Toc40777409)

[7.1 负载/压力测试 198](#_Toc40777410)

[7.2 效能测试 198](#_Toc40777411)

[7.3 软件辅助功能测试 198](#_Toc40777412)

[7.4 全球化测试 198](#_Toc40777413)

[7.5 兼容性测试 198](#_Toc40777414)

# 引言

## 目的

**本文档主要用于分析软件工程实验I组的产品测试，即“对Node-RED工具的消息聚集和图形可视化拓展”的相关测试，并且科学合理地设计相关的测试用例，完成对预期软件需求的拓展功能和非功能测试。**

## **软件测试的主要内容**

软件测试是一种实际输出与预期输出间的审核或者比较过程。软件测试的经典定义是：在规定的条件下对程序进行操作，以发现程序错误，衡量软件质量，并对其是否能满足设计要求进行评估的过程。本次软件测试阶段的主要工作如下：

总结项目实现的内容，结合需求分析阶段的内容，分析测试需求，编写测试计划及测试规格说明书。具体来说，为：

1) 编写有效的、覆盖面广的测试用例；

2) 研究相关测试技术；

3) 按计划实施测试工作，提交测试报告。

## 文档描述

本次测试需求规格说明书将参照《需求规格说明书v2.6.1》及已经实现的项目内容，给出需求用例与测试用例的对照表。对于每个测试用例，需先给出测试策略的描述，然后按照相关标准对测试用例进行说明。由于本项目是基于Node-RED平台实现的，项目的功能需求是针对于该平台的需求，在测试中无需纳入考虑范围，因此本环节只需完成拓展需求以及非功能需求两个方面的测试工作。

## 测试需求与需求对照表

本次测试严格遵照《需求规格说明书》以及项目实现内容来进行设计与实现，其中，测试需求与需求的对照表如表1所示。

表1测试需求与需求对照表

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 测试需求编号 | 需求类别 | 与需求规格说明书对应需求项 | 测试需求内容 | 测试方面 | 测试用例 | 备注 |
| TR1.1 | 功能需求 | 3.2.1 | 创建流程 |  |  | Node-RED平台功能，无需单独测试 |
| TR1.2 | 3.2.2 | 安装拓展包 |  |  |
| TR1.3 | 3.2.3 | 部署流程 |  |  |
| TR1.4 | 3.2.4 | 触发事件 |  |  |
| TR1.5 | 3.2.5 | 监控流程 |  |  |
| TR1.6 | 3.2.6 | 开发拓展包 |  |  |
| TR1.7 | 3.2.7 | 发布拓展包 |  |  |
| TR2.1 | 拓展需求 | 3.3.4 | 创建飞书激活节点 | 节点功能 | 飞书激活节点测试 |  |
| TR2.2 | 3.3.1 | 创建飞书配置节点 | 节点功能 | feishu-config节点功能测试 |  |
| TR2.3 | 3.3.2 | 创建飞书获得群列表节点 | 节点加载 | feishu-all-group  节点加载测试 |  |
| 节点功能 | feishu-all-group  节点功能测试 |  |
| TR2.4 | 3.3.3 | 创建飞书获得群信息节点 | 节点功能 | feishu-fetch-group-info  节点功能测试 |  |
| TR2.5 | 3.3.5 | 创建飞书获得接收消息节点 | 节点子功能 | feishu-msg  节点接收私聊文本测试 |  |
| 节点子功能 | feishu-msg  节点接收私聊图片测试 |  |
| 节点子功能 | feishu-msg  节点接收群聊文本测试 |  |
| 节点子功能 | feishu-msg  节点接收群聊图片测试 |  |
| TR2.6 | 3.3.6 | 创建飞书获得上传图片节点 | 节点加载 | feishu-upload-image  节点加载测试 |  |
| 节点功能 | feishu-upload-image  节点上传图片测试 |  |
| TR2.7 | 3.3.7 | 创建飞书获取图片节点 | 节点加载 | feishu-fetch-image  节点加载测试 |  |
| 节点功能 | feishu-fetch-image  节点获取图片测试 |  |
| TR2.8 | 3.3.8 | 创建飞书添加目标节点 | 节点部署 | 飞书添加目标节点部署测试 |  |
| 节点子功能 | 飞书添加目标节点正则表达式筛选器测试 |  |
| 节点子功能 | 飞书添加目标节点列表筛选器测试 |  |
| 节点子功能 | 飞书添加目标节点筛选对象及属性测试 |  |
| 节点子功能 | 飞书添加目标节点直接添加模式测试 |  |
| 节点初始化 | 飞书添加目标节点初始化测试 |  |
| 异常处理 | 飞书添加目标节点重复添加测试 |  |
| 异常处理 | 飞书添加目标节点异常测试 |  |
| 异常处理 | 飞书添加目标节点配置异常测试 |  |
| TR2.9 | 3.3.9 | 创建飞书移除目标节点 | 节点部署 | 飞书移除目标节点部署测试 |  |
| 节点子功能 | 飞书移除目标节点正则表达式筛选器测试 |  |
| 节点子功能 | 飞书移除目标节点列表筛选器测试 |  |
| 节点子功能 | 飞书移除目标节点筛选对象及属性测试 |  |
| 节点初始化 | 飞书移除目标节点初始化测试 |  |
| 异常处理 | 飞书移除目标节点异常测试 |  |
| 异常处理 | 飞书移除目标节点配置异常测试 |  |
| TR2.10 | 3.3.10 | 创建飞书发送消息节点 | 节点加载 | feishu-send  节点加载测试 |  |
| 节点子功能 | feishu-send  节点发送群聊图片测试 |  |
| 节点子功能 | feishu-send  节点发送群聊文本测试 |  |
| 节点子功能 | feishu-send  节点发送私聊图片测试 |  |
| 节点子功能 | feishu-send  节点发送私聊文本测试 |  |
| TR2.11 | 3.3.11 | 创建图表渲染节点 | 节点部署 | chart-render节点部署测试 |  |
| 节点子功能 | chart-render节点Line chart绘制测试 |  |
| 节点子功能 | chart-render节点Bar chart绘制测试 |  |
| 节点子功能 | chart-render节点Pie chart绘制测试 |  |
| 节点子功能 | chart-render节点Radar chart绘制测试 |  |
| 节点子功能 | chart-render节点Polar area chart绘制测试 |  |
| 节点子功能 | chart-render节点修改图表长宽测试 |  |
| 节点子功能 | chart-render节点数据窗口测试 |  |
| 节点子功能 | chart-render节点binary输出测试 |  |
| TR2.12 | 3.3.12 | 创建模板渲染节点 | 节点部署 | 模板渲染节点部署测试 |  |
| 节点功能 | 模板渲染节点功能正确性测试 |  |
| 异常处理 | 模板渲染节点异常处理测试 |  |
| 异常处理 | 模板渲染节点忽略空值输入测试 |  |
| 节点子功能 | 模板渲染节点输出时清空字段功能正确性测试 |  |
| 节点子功能 | 模板渲染节点输出后不清空字段功能正确性测试 |  |
| 节点子功能 | 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 |  |
| 节点子功能 | 模板渲染节点不等待所有字段功能正确性测试 |  |
| TR2.13 | 3.3.13 | 创建窗口节点 | 节点部署 | 窗口节点部署测试 |  |
| 异常处理 | 窗口节点配置异常处理测试 |  |
| 节点子功能 | 窗口节点session模式功能正确性测试 |  |
| 节点子功能 | 窗口节点session模式首次输入激活测试 |  |
| 异常处理 | 窗口节点session模式配置异常处理测试 |  |
| 节点子功能 | 窗口节点tumbling时间模式功能正确性测试 |  |
| 节点子功能 | 窗口节点tumbling时间模式无数据测试 |  |
| 异常处理 | 窗口节点tumbling时间模式配置异常处理测试 |  |
| 节点子功能 | 窗口节点tumbling数据模式功能正确性测试 |  |
| 异常测试 | 窗口节点tumbling数量模式配置异常处理测试 |  |
| 节点子功能 | 窗口节点sliding时间模式功能正确性测试 |  |
| 异常处理 | 窗口节点sliding时间模式窗口时长配置异常处理测试 |  |
| 异常处理 | 窗口节点sliding时间模式滑动时长配置异常处理测试 |  |
| 节点子功能 | 窗口节点sliding数量模式功能正确性测试 |  |
| 异常处理 | 窗口节点sliding数量模式窗口数量配置异常处理测试 |  |
| 异常处理 | 窗口节点sliding数量模式滑动数量配置异常处理测试 |  |
| TR4.5 | 非功能性需求 | 3.4.1 | 兼容性测试 |  |  |  |
| TR4.6 | 3.4.2 | 负载/压力测试 |  |  | 由于飞书平台的限制，无法进行测试 |
| TR4.5 | 3.4.3 | 效能测试 |  |  | 由于飞书平台的限制，无法进行测试 |

# 测试计划

## 测试工具介绍

* + 1. 单元测试框架

本项目组计划使用mocha作为我们的单元测试框架。mocha是一个适用于Node.js程序的JavaScript测试框架，支持异步测试、断言库的使用等。运行界面示意如图1所示。

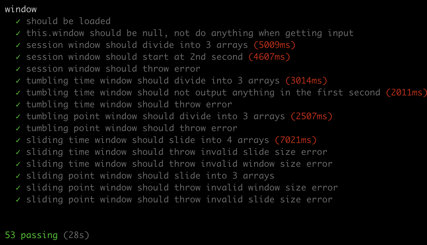


图1 mocha运行界面示意

* + 1. 覆盖率工具

本项目组选择采用nyc代码覆盖率工具配合mocha测试框架来检测当前的测试覆盖率。nyc有四个测量维度，分别是：

• line coverage

• function coverage

• branch coverage

• statement coverage

借助于nyc，测试人员可以根据当前覆盖率进行增强测试。运行界面示意如图2所示。

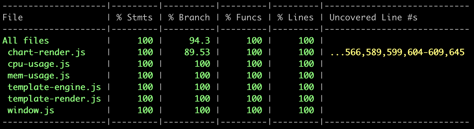


图2 nyc运行界面示意

* + 1. HTTP服务器的模拟期望库

由于项目中部分节点（飞书相关的节点）涉及到向飞书发送HTTP请求的功能，所以需要有相应的模拟测试，测试人员选择了nock，nock可以拦截所有Node.js的HTTP请求，以检查HTTP请求的正确性。nock示意代码如图3所示。



图3 nock代码示意图

* + 1. Node-Red测试框架

[Node Test Helper](https://github.com/node-red/node-red-node-test-helper)是Node-Red提供的测试框架，其配合mocha可以对节点进行功能测试以及集成测试。

## 测试类型

* + 1. 白盒测试

白盒测试主要用于配合黑盒测试使用，其设计思路在于增添黑盒测试用例以增强代码覆盖率。这部分工作将由负责开发的同学完成，其中每个人根据各自的开发模块的内部实现来设计测试用例。包括部分单元测试以及冒烟测试。

* + 1. 黑盒测试

黑盒测试用于大部分的测试用例设计，包括单元测试以及场景测试。

## 测试目的

本项目组的测试属于验证构建测试，以及验收测试。

## 测试方法

该测试整体属于伙伴测试，大部分测试用例是由开发人员自己设计开发，本地通过测试后才提交代码。

测试中包含随机测试，场景测试部分属于随机测试。

同时，该测试还可以作为回归测试，在后续功能迭代时，需要重新运行所有测试用例，避免新版本的功能退化。

# 拓展功能测试

在该部分项目组将借助mocha单元测试框架实现各个模块的自动化单元测试。然后借助Node-Red的测试框架Node Test Helper，该框架支持对节点功能和节点间的依赖关系进行验证，帮助实现各个模块的自动化功能测试和集成测试。

## 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 | |  |
| 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 | |  | | 检查tenant\_token是否与预设值相同 | |  |
| 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 | |  | | 检查tenant\_token是否与预设值相同 | |  |
|  | feishu-all-group节点没有报错 | |  | | 检查是否有未处理的异常 | |  |
| 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feishu-test-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 | |  | | 检查tenant\_token是否与预设值相同 | |  |
|  | feishu-all-group节点没有报错 | |  | | 检查是否有未处理的异常 | |  |
| 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 | |  | | 检查返回消息的群列表是否与测试群列表相同 | |  |
| 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feishu-test-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 | |  | test | |  | | 内容一致 | |  |
| 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 | |  | | 检查tenant\_token是否与预设值相同 | |  |
| 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 | |  | | 检查返回消息的群信息是否与测试群信息相同 | |  |
| 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test-helper节点的工作流 | |  |  | |  | |  | |  |
| 3 | 设置test-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 | |  | | 返回的消息与预期是否相等 | |  |
| 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test-helper节点的工作流 | |  |  | |  | |  | |  |
| 3 | 设置test-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 | |  | | 返回的消息与预期是否相等 | |  |
| 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test-helper节点的工作流 | |  |  | |  | |  | |  |
| 3 | 设置test-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 | |  | | 返回的消息与预期是否相等 | |  |
| 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test-helper节点的工作流 | |  |  | |  | |  | |  |
| 3 | 设置test-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 | |  | | 返回的消息与预期是否相等 | |  |
| 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 | |  | | 检查tenant\_token是否与预设值相同 | |  |
|  | feishu-upload-image节点没有报错 | |  | | 检查是否有未处理的异常 | |  |
| 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 | |  | | 检查tenant\_token是否与预设值相同 | |  |
|  | 拦截器收到合法请求 | |  | | 上传的Buffer是否与预设的相同 | |  |
|  | Helper收到预设的TEST\_IMAGE\_KEY | |  | | 返回的image\_key与预设是否一致 | |  |
| 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 | |  | | 检查tenant\_token是否与预设值相同 | |  |
|  | feishu-fetch-image节点没有报错 | |  | | 检查是否有未处理的异常 | |  |
| 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feishu-test-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 | |  | | 检查tenant\_token是否与预设值相同 | |  |
| 预设的image\_key | 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 | |  |
| 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 | |  | | 内容一致 | |  |
| 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" | |  | | 同预期结果一致 | |  |
| 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\_1" | |  | | 同预期结果一致 | |  |
| 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\_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" | |  | | 同预期结果一致 | |  |
| 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" | |  | | 同预期结果一致 | |  |
| 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"} | |  | | 同预期结果一致 | |  |
| 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 | |  | | 同预期结果一致 | |  |
| 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"} | |  | | 同预期结果一致 | |  |
| 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" | |  | | 同预期结果一致 | |  |
| 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" | |  | | 同预期结果一致 | |  |
| 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" | |  | | 同预期结果一致 | |  |
| 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 | |  | | 同预期结果一致 | |  |
| 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 | |  | | 同预期结果一致 | |  |
| 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 | |  | | 同预期结果一致 | |  |
| 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 | |  | | 同预期结果一致 | |  |
| 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 | |  | | 同预期结果一致 | |  |
| 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 | |  | | 同预期结果一致 | |  |
| 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.” | |  | | 同预期结果一致 | |  |
| 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.” | |  | | 同预期结果一致 | |  |
| 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” | |  | | 同预期结果一致 | |  |
| 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” | |  | | 同预期结果一致 | |  |
| 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” | |  | | 同预期结果一致 | |  |
| 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” | |  | | 同预期结果一致 | |  |
| 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” | |  | | 同预期结果一致 | |  |
| 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” | |  | | 同预期结果一致 | |  |
| 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 | |  | | 同预期结果一致 | |  |
| 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 | |  | | 同预期结果一致 | |  |
| 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 | |  | | 同预期结果一致 | |  |
| 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 | |  | | 同预期结果一致 | |  |
| 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 | |  | | 同预期结果一致 | |  |
| 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 | |  | | 同预期结果一致 | |  |
| 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 | |  | | 内容一致 | |  |
| 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1；feishu\_meta\_info.target.chat[0]为"test\_chat\_id\_2" | |  | | 同预期结果一致 | |  |
| 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" | |  | | 同预期结果一致 | |  |
| 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\_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" | |  | | 同预期结果一致 | |  |
| 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" | |  | | 同预期结果一致 | |  |
| 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"} | |  | | 同预期结果一致 | |  |
| 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" | |  | | 同预期结果一致 | |  |
| 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"} | |  | | 同预期结果一致 | |  |
| 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" | |  | | 同预期结果一致 | |  |
| 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 | |  | | 同预期结果一致 | |  |
| 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 | |  | | 同预期结果一致 | |  |
| 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 | |  | | 同预期结果一致 | |  |
| 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 | |  | | 同预期结果一致 | |  |
| 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” | |  | | 同预期结果一致 | |  |
| 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” | |  | | 同预期结果一致 | |  |
| 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” | |  | | 同预期结果一致 | |  |
| 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 | |  | | 同预期结果一致 | |  |
| 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 | |  | | 同预期结果一致 | |  |
| 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 | |  | | 同预期结果一致 | |  |
| 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 | |  | | 同预期结果一致 | |  |
| 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 | |  | | 检查tenant\_token是否与预设值相同 | |  |
|  | feishu-send节点没有报错 | |  | | 检查是否有未处理的异常 | |  |
| 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 | |  | | 检查是否与预设的token一致 | |  |
| 测试消息1 | 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 | |  |
| 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 | |  | | 检查是否与预设的token一致 | |  |
| 测试消息1 | 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 | |  |
| 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 | |  | | 检查是否与预设的token一致 | |  |
| 测试消息1 | 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 | |  |
| 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 | |  | | 检查是否与预设的token一致 | |  |
| 测试消息1 | 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 | |  |
| 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  }  } | | | | | | | | |

## 创建并配置图表渲染节点测试

* + 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 | 创建一个含有无配置的chart-render节点的工作流，name属性为test | |  |  | |  | |  | |  |
| 3 | 部署工作流 | |  |  | |  | |  | |  |
| 4 | 获取chart-render节点的name属性 | |  | test | |  | | 内容一致 | |  |
| 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/${TESTID}\_6.1.jpg | |  | | 与gt文件夹中的同名图片肉眼比对一致 | |  |
| 依次输入两个number类型的消息：   1. {payload: 123}   {payload:123} | helper节点收到两次数据，将保存到output/${TESTID}\_6.2.1.jpg和output/${TESTID}\_6.2.2.jpg | |  | | 与gt文件夹中的同名图片肉眼比对一致 | |  |
| 依次输入两个Array类型的消息：   1. {payload: [1, 2, 3]}   {payload: [1, 2, 3]} | helper节点收到两次数据，将保存到output、${TESTID}\_6.3.1.jpg和output/${TESTID}\_6.3.2.jpg | |  | | 与gt文件夹中的同名图片肉眼比对一致 | |  |
| 输入非法类型的消息（以bool类型为例）：  {payload: false} | chart-render节点抛出异常被node-red框架捕获，helper节点不会收到消息 | |  | | chart-render节点没有输出 | |  |
| 输入带横坐标的number信息（并包含横坐标格式化字符串）：  {payload: "null", x: "2020-10-1", y: 123, format: "date:%Y-%m-%d"} | helper节点收到数据保存到output/${TESTID}\_6.4.jpg | |  | | 与gt文件夹中的同名图片肉眼比对一致 | |  |
| 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/${TESTID}\_6.5.jpg | |  | | 与gt文件夹中的同名图片肉眼比对一致 | |  |
| 输入带横坐标的错误格式信息：  {payload: "null", x: false, y: false} | chart-render节点抛出异常被node-red框架捕获，helper节点不会收到消息 | |  | | chart-render节点没有输出 | |  |
| 输入带config的信息，以修改节点的配置：  { payload: 123, config: { chartType: "Line chart", title: "", xLabel: "", yLabel: "", dataWindow: 0, output: "base64", height: 200, width: 400}} | helper节点收到数据保存到output/${TESTID}\_6.7.jpg | |  | | 与gt文件夹中的同名图片肉眼比对一致 | |  |
| 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 | |  | | 与gt文件夹中的同名图片肉眼比对一致 | |  |
| 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-${TEXTID}\_5.1>} | |  | | 与gt文件夹中的同名图片肉眼比对一致 | |  |
| 输入带横坐标的信息（并包含横坐标格式化字符串）：  {payload: "null", x: "2020-10-1", y: 123, format: "date:%Y-%m-%d"} | helper节点收到数据保存到output/${TESTID}\_5.2.jpg | |  | | 与gt文件夹中的同名图片肉眼比对一致 | |  |
| 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/${TESTID}\_5.1.jpg | |  | | 与gt文件夹中的同名图片肉眼比对一致 | |  |
| 输入带横坐标的信息（并包含横坐标格式化字符串）：  {payload: "null", x: "2020-10-1", y: 123, format: "date:%Y-%m-%d"} | helper节点收到数据保存到output/${TESTID}\_5.2.jpg | |  | | 与gt文件夹中的同名图片肉眼比对一致 | |  |
| 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/${TESTID}\_5.1,jpg | |  | | 与gt文件夹中的同名图片肉眼比对一致 | |  |
| 输入带横坐标的信息（并包含横坐标格式化字符串）：  {payload: "null", x: "2020-10-1", y: 123, format: "date:%Y-%m-%d"} | helper节点收到数据保存到output/${TESTID}\_5.2.jpg | |  | | 与gt文件夹中的同名图片肉眼比对一致 | |  |
| 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/${TESTID}\_5.1.jpg | |  | | 与gt文件夹中的同名图片肉眼比对一致 | |  |
| 输入带横坐标的信息（并包含横坐标格式化字符串）：  {payload: "null", x: "2020-10-1", y: 123, format: "date:%Y-%m-%d"} | helper节点收到数据，保存到output/${TESTID}\_5.2.jpg | |  | | 与gt文件夹中的同名图片肉眼比对一致 | |  |
| 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 | |  | | 与gt文件夹中的同名图片肉眼比对一致 | |  |
| 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}   {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 | |  | | 与gt文件夹中的同名图片肉眼比对一致 | |  |
| 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 | |  | | 与gt文件夹中的同名图片肉眼比对一致 | |  |
| 6 | 停止node-red测试环境 | |  |  | |  | |  | |  |
| **备注** | |  | | | | | | | | |

## 创建并配置模板渲染节点测试

* + 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** | | 测试模板窗口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 | 创建含有一个无配置的模板节点的工作流，name属性为template-render | |  |  | |  | |  | |  |
| 3 | 部署工作流 | |  |  | |  | |  | |  |
| 4 | 获取窗口节点的name属性 | |  | template-render | |  | | 数据一致 | |  |
| 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:"hhttp://tete.com/imi"} | 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 | |  |
| 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" | |  | | 报错，且报错信息一致 | |  |
| 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) " | |  | | 只有在本次输入才有输出，且数据一致 | |  |
| 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) " | |  | | 本次输入有输出，且数据一致 | |  |
| 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) " | |  | | 本次输有输出，且数据一致 | |  |
| 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)" | |  | | 本次输入有输出，且数据一致 | |  |
| 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)" | |  | | 本次输入有输出，且数据一致 | |  |
| 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 ![]({{ }})" | |  | | 数据一致 | |  |
| 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}})" | |  | | 数据一致 | |  |
| 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)" | |  | | 数据一致 | |  |
| 7 | 停止node-red测试环境 | |  |  | |  | |  | |  |
| **备注** | | 无 | | | | | | | | |

## 创建并配置窗口节点测试

* + 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 | 创建含有一个无配置的窗口节点的工作流，name属性为window | |  |  | |  | |  | |  |
| 3 | 部署工作流 | |  |  | |  | |  | |  |
| 4 | 获取窗口节点的name属性 | |  | window | |  | | 数据一致 | |  |
| 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，对每个输入都报错 | | | | | | | | |
| **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" | |  | | 报错且错误信息一致 | |  |
| 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]} | |  | | 时间点正确，数据一致 | |  |
| 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]} | |  | | 时间点正确，数据一致 | |  |
| 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]} | |  | | 时间点正确，数据一致 | |  |
| 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]} | |  | | 时间点正确，数据一致 | |  |
| 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]} | |  | | 时间点正确，数据一致 | |  |
| 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" | |  | | 报错且错误信息一致 | |  |
| 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]} | |  | | 时间点正确，数据一致 | |  |
| 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]} | |  | | 时间点正确，数据一致 | |  |
| 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]} | |  | | 时间点正确，数据一致 | |  |
| 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]} | |  | | 时间点正确，数据一致 | |  |
| 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" | |  | | 报错且错误信息一致 | |  |
| 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]} | |  | | 输入后立刻收到消息，数据一致 | |  |
| 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]} | |  | | 输入后立刻收到消息，数据一致 | |  |
| 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]} | |  | | 输入后立刻收到消息，数据一致 | |  |
| 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" | |  | | 报错且错误信息一致 | |  |
| 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]} | |  | | 时间点正确，数据一致 | |  |
| 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]} | |  | | 时间点正确，数据一致 | |  |
| 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]} | |  | | 时间点正确，数据一致 | |  |
| 13 | 在t=6000ms时 | |  | helper收到数据{payload:[0,0]} | |  | | 时间点正确，数据一致 | |  |
| 14 | 在t=7000ms时 | |  | helper收到数据{payload:[0]} | |  | | 时间点正确，数据一致 | |  |
| 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" | |  | | 报错且错误信息一致 | |  |
| 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" | |  | | 报错且错误信息一致 | |  |
| 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]} | |  | | 输入后立刻收到输出，数据一致 | |  |
| 8 | 向tumbling时间窗口节点输入数据 | | {payload:0} | helper收到数据{payload:[0,0,0]} | |  | | 输入后立刻收到输出，数据一致 | |  |
| 9 | 向tumbling时间窗口节点输入数据 | | {payload:0} | helper收到数据{payload:[0,0,0]} | |  | | 输入后立刻收到输出，数据一致 | |  |
| 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" | |  | | 报错且错误信息一致 | |  |
| 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" | |  | | 报错且错误信息一致 | |  |
| 4 | 停止node-red测试环境 | |  |  | |  | |  | |  |
| **备注** | |  | | | | | | | | |

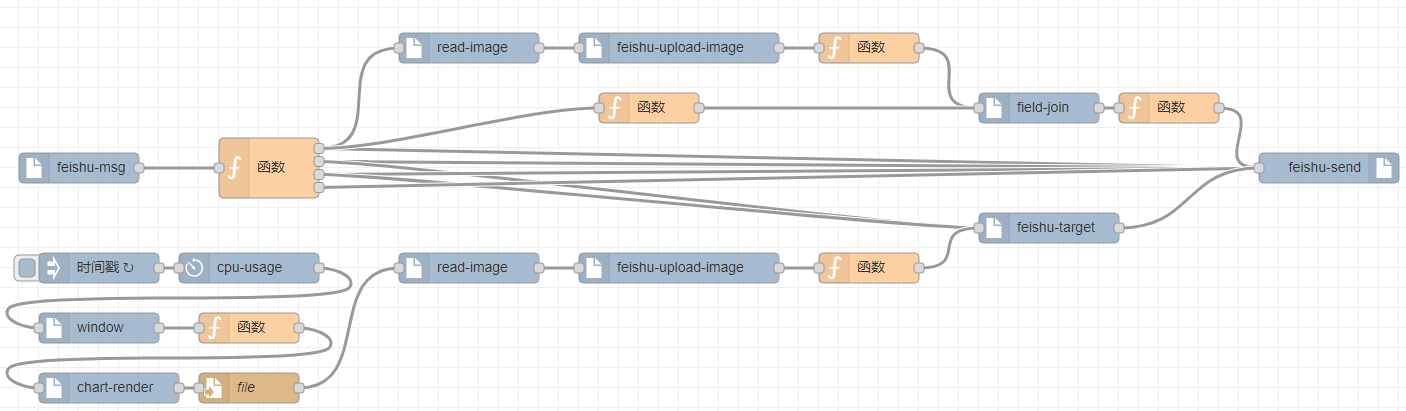
## 创建获取CPU使用率节点测试

* + 1. 获取CPU使用率测试

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **测试用例名称** | | 获取CPU使用率测试 | **测试用例标识** | | TC068 | | **测试需求标识** | |  | |
| **简要描述** | | 测试CPU使用率节点能够以百分比返回CPU使用率 | | | | | | | | |
| **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CPU使用率节点和一个helper节点的工作流 | |  |  | |  | |  | |  |
| 3 | 部署工作流 | |  |  | |  | |  | |  |
| 4 | 向CPU使用率节点输入数据 | | {} | helper收到CPU使用率，且该值取值范围为[0,1] | |  | | 数据在范围内 | |  |
| 5 | 停止node-red测试环境 | |  |  | |  | |  | |  |
| **备注** | | 此节点的开发主要用于搭建示例场景。 | | | | | | | | |

# 场景测试

**在场景测试中，测试人员将多个开发的功能节点串联为一个完整功能，提供了从飞书机器人获取node-red服务器当前CPU使用情况图和订阅CPU使用情况图功能。该功能的流程如图4所示。其中使用到了开发人员开发的feishu-msg，read-image，feishu-upload-image，field-join，feishu-send，cpu-usage，window节点。**



**图4 通过飞书获取CPU使用情况流程图**

**该场景的测试用例将在本章说明。**

## 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** | | TC069 | | **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 | |  | | 返回结果 | |  |
| **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** | | TC070 | | **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 | |  | | 返回提示 | |  |
| 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 | |  | | 正确周期性收到图片 | |  |
| **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** | | TC071 | | **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 | |  | | 返回提示 | |  |
| 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 | |  | | 正确周期性收到图片 | |  |
| 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 | |  | | 返回提示 | |  |
| 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 | |  | | 不再收到订阅 | |  |
| **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** | | TC072 | | **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 | |  | | 能够容错，正常返回提示 | |  |
| **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** | | TC073 | | **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 | |  | | 返回提示 | |  |
| 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 | |  | | 正确周期性收到图片 | |  |
| 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 | |  | | 返回结果 | |  |
| 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 | |  | | 返回提示 | |  |
| 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 | |  | | 不再收到订阅 | |  |
| **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** | | TC074 | | **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 | |  | | 返回消息格式不正确提示信息 | |  |
| **备注** | | 输入与结果为与飞书机器人私聊的消息 | | | | | | | | |

# 系统测试

**无，本项目开发内容不涉及系统层面。**

# AB测试

**无，本项目开发内容不涉及AB环境测试。**

# 非功能需求测试

## 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7.1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的在不同操作系统及浏览器中部署运行。