

Message-Pusher 接口配置规则

本文档详细说明了如何在 `message-pusher` 中配置 webhook 接口，以接收来自 `rss2tg` 的推送消息。

目录

- 配置概述
- 数据结构说明
- 配置方案
 - 方案一：基础配置（推荐）
 - 方案二：双字段配置
 - 方案三：链接预览优化配置
 - 方案四：自定义格式配置
- 重要限制说明
- 配置步骤
- 故障排除

配置概述

`rss2tg` 通过 webhook 向 `message-pusher` 发送 JSON 格式的消息数据。在 `message-pusher` 中需要配置：

- 提取规则**：定义从 JSON 数据中提取哪些字段
- 构建规则**：定义如何使用提取的字段构建最终消息

数据结构说明

`rss2tg` 发送给 `message-pusher` 的 JSON 数据结构：

```
{
  "title": "文章标题",
  "description": "分组: 技术资讯 | 关键词: VPS, 优惠 | 时间: 2024-01-20 15:30:45",
  "content": "### 📖 【技术资讯】RSS推送\n\n**标题: ** 文章标题\n\nhttps://example.com/article\n\n**关键词: ** #VPS #优惠\n\n**时间: ** 2024-01-20 15:30:45",
  "url": "https://example.com/article",
  "group": "技术资讯",
  "keywords": "VPS, 优惠",
  "timestamp": "2024-01-20 15:30:45"
}
```

字段说明

字段	类型	说明	示例
title	字符串	RSS 文章标题	"最新VPS优惠活动"
description	字符串	简短描述, 包含分组、关键词、时间	"分组: 技术资讯 关键词: VPS, 优惠 时间: 2024-01-20 15:30:45"
content	字符串	完整的 Markdown 格式内容	"### 📄 【技术资讯】RSS推送..."
url	字符串	文章链接	"https://example.com/article"
group	字符串	RSS 分组名称	"技术资讯"
keywords	字符串	匹配的关键词, 逗号分隔	"VPS, 优惠"
timestamp	字符串	发布时间 (中国时区)	"2024-01-20 15:30:45"

配置方案

方案一：基础配置（推荐）

适用场景：大多数用户，简单易用，显示效果好

提取规则：

```
{
  "title": "title",
  "description": "description",
  "content": "content",
  "url": "url",
  "group": "group",
  "keywords": "keywords",
  "timestamp": "timestamp"
}
```

构建规则：

```
{
  "content": "$content"
}
```

效果预览：



📄 【技术资讯】RSS推送

标题: 最新VPS优惠活动

https://example.com/article

关键词: #VPS #优惠

时间: 2024-01-20 15:30:45

优点:

- ☑ 配置简单, 直接使用预格式化的内容
- ☑ 支持 Markdown 格式, 显示效果好
- ☑ 链接单独一行, 便于生成预览
- ☑ 包含所有必要信息

方案二: 双字段配置

适用场景: 需要同时显示简短描述和详细内容

提取规则:

```
{
  "title": "title",
  "description": "description",
  "content": "content",
  "url": "url",
  "group": "group",
  "keywords": "keywords",
  "timestamp": "timestamp"
}
```

构建规则:

```
{
  "description": "$description",
  "content": "$content"
}
```

效果预览:

分组: 技术资讯 | 关键词: VPS, 优惠 | 时间: 2024-01-20 15:30:45

📄 【技术资讯】RSS推送

标题: 最新VPS优惠活动

https://example.com/article

关键词: #VPS #优惠

时间: 2024-01-20 15:30:45

优点:

- ☒ 提供简短描述和详细内容
- ☒ 适合需要快速浏览的场景

方案三: 链接预览优化配置

适用场景: 重点关注链接预览效果

提取规则:

```
{
  "title": "title",
  "description": "description",
  "content": "content",
  "url": "url",
  "group": "group",
  "keywords": "keywords",
  "timestamp": "timestamp"
}
```

构建规则:

```
{
  "content": "📄 **$title**\n\n🔗 分组: $group\n\n🌐 $url\n\n🔍 关键词: $keywords\n\n🕒 时间: $timestamp"
}
```

效果预览:

📄 **最新VPS优惠活动**

🔗 分组: 技术资讯

https://example.com/article

🔍 关键词: VPS, 优惠

🕒 时间: 2024-01-20 15:30:45

优点:

- ☒ 链接突出显示, 便于预览
- ☒ 格式简洁清晰
- ☒ 适合移动端查看

方案四: 自定义格式配置

适用场景: 需要特定格式或样式的用户

提取规则:

```
{
  "title": "title",
  "description": "description",
  "content": "content",
  "url": "url",
  "group": "group",
  "keywords": "keywords",
  "timestamp": "timestamp"
}
```

构建规则:

```
{
  "content": "🔔 RSS 推送通知\n\n标题: $title\n分组: $group\n链接: $url\n关键词: $keywords\n时间: $timestamp"
}
```

效果预览:

🔔 RSS 推送通知

标题: 最新VPS优惠活动
分组: 技术资讯
链接: https://example.com/article
关键词: VPS, 优惠
时间: 2024-01-20 15:30:45

优点:

- ☒ 完全自定义格式

- ☒ 可根据需求调整样式
- ☒ 适合特殊显示需求

重要限制说明

⚠ **关键限制：**根据 message-pusher 的实现限制，在构建规则中：

1. 只有 **content** 和 **description** 字段有效
2. 其他字段（如 **title**、**url**、**group** 等）在构建规则中无效，会导致数据为空
3. 必须在构建规则中使用 **\$变量名** 格式引用提取的数据

✖ 错误配置示例

```
{
  "title": "$title",
  "url": "$url",
  "group": "$group"
}
```

这种配置会导致 title、url、group 字段为空

☒ 正确配置示例

```
{
  "content": "标题: $title\n链接: $url\n分组: $group"
}
```

所有信息都包含在 content 字段中

配置步骤

1. 登录 message-pusher 后台

访问你的 message-pusher 管理界面。

2. 创建 webhook 通道

1. 进入"产品配置" -> "webhook 配置"
2. 点击"新建 webhook 通道"
3. 填写通道名称（如：rss2tg-webhook）

3. 配置提取规则

在"提取规则"中输入以下 JSON（推荐使用方案一）：

```
{
  "title": "title",
  "description": "description",
  "content": "content",
  "url": "url",
  "group": "group",
  "keywords": "keywords",
  "timestamp": "timestamp"
}
```

4. 配置构建规则

在"构建规则"中输入以下 JSON（推荐使用方案一）：

```
{
  "content": "$content"
}
```

5. 保存并获取 webhook URL

保存配置后，复制生成的 webhook URL，格式类似：

```
http://your-domain:3000/webhook/your_webhook_id
```

6. 配置 rss2tg

将获取的 webhook URL 配置到 rss2tg 中：

环境变量方式：

```
WEBHOOK_ENABLED=true
WEBHOOK_URL=http://your-domain:3000/webhook/your_webhook_id
```

配置文件方式：

```
webhook:
  enabled: true
  url: "http://your-domain:3000/webhook/your_webhook_id"
  timeout: 10
  retry_count: 3
```

故障排除

问题1：消息内容为空

原因：构建规则中使用了无效字段

解决方案：

- 检查构建规则是否只使用了 `content` 和 `description` 字段
- 确保使用 `$变量名` 格式引用数据

问题2：变量未替换

原因：提取规则中的字段名与 rss2tg 发送的数据不匹配

解决方案：

- 确保提取规则中的字段名与数据结构完全一致
- 检查 JSON 格式是否正确

问题3：链接预览不显示

原因：链接被其他文本包围，影响预览生成

解决方案：

- 使用方案一或方案三，将链接单独放在一行
- 避免在链接前后添加过多文本

问题4：格式显示异常

原因：Markdown 格式不正确或平台不支持

解决方案：

- 检查 Markdown 语法是否正确
- 根据目标平台调整格式（如去掉 Markdown 标记）

问题5：webhook 推送失败

原因：网络连接问题或 message-pusher 服务异常

解决方案：

- 检查 message-pusher 服务是否正常运行
- 确认 webhook URL 是否正确
- 查看 rss2tg 日志获取详细错误信息

测试配置

配置完成后，可以通过以下方式测试：

1. 查看 rss2tg 日志：
-


```
docker logs rss2tg
```

2. **检查 message-pusher 日志：** 查看是否收到 webhook 请求
3. **手动触发测试：** 在 rss2tg 中添加一个测试 RSS 源，观察推送效果

相关链接

- [message-pusher 官方文档](#)
- [rss2tg 项目主页](#)
- [返回主文档](#)