RSS to Telegram Bot 使用文档

English | 简体中文

🏷 简介

简称 rss2tg，用于将自定义 RSS 地址，字段，刷新时间里的相关帖子即时发送到自定义 TG 用户或频道，省去你刷帖子的时间

支持 AMD64/ARM64

镜像大小 17M，内存占用 10M

——By drfyup

1. 部署方法

1.1 使用 Docker Compose（推荐）

1. 确保已安装 Docker 和 Docker Compose（方法自寻）。
2. 克隆或下载项目代码到本地。

```
git clone https://github.com/3377/rss2tg.git
```

3. 进入项目目录。
4. 编辑 `docker-compose.yml` 文件，修改环境变量：

-- 进入任意目录或直接当前目录，新建 docker-compose.yml 文件，填入以下内容

```
version: "3"
services:
  rss2tg:
    container_name: rss2tg
    image: drfyup/rss2tg:latest
    volumes:
      - ./config/config.yaml:/app/config/config.yaml
      - ./data:/app/data
    environment:
      - TELEGRAM_BOT_TOKEN=your_bot_token_here
      - TELEGRAM_USERS=user_id_1,user_id_2
      - TELEGRAM_CHANNELS=@channel_1,@channel_2
      - TZ=Asia/Shanghai
    restart: unless-stopped
```

将`your_bot_token_here` 替换为您的 Telegram Bot Token，`user_id_1,user_id_2` 替换为您要接收消息的用户 ID，`@channel_1,@channel_2` 替换为您要发送消息的频道名称。

5. 运行以下命令启动容器：

```
docker-compose up  -d
```

1.2 使用 Docker Run

1. 构建 Docker 镜像：

```
docker pull drfyup/rss2tg:latest
```

2. 运行 Docker 容器：

```
docker run -d \
--name rss2tg \
-v $(pwd)/config:/app/config \
-v $(pwd)/data:/app/data \
-e TELEGRAM_BOT_TOKEN=your_bot_token_here \
-e TELEGRAM_USERS=user_id_1,user_id_2 \
-e TELEGRAM_CHANNELS=@channel_1,@channel_2 \
-e TELEGRAM_API_URL=http://xxx.deno.dev/telegram \
-e TZ=Asia/Shanghai \
--restart unless-stopped \
drfyup/rss2tg:latest
```

请替换环境变量中的相应值。

2. 程序使用说明

2.1 配置文件

程序支持通过 YAML 配置文件或环境变量进行配置。配置文件位于 `/app/config/config.yaml`。如果该文件不存在，程序将使用环境变量进行初始配置。 环境变量读取优先级高于配置文件。

配置文件示例：

```
telegram:
  bot_token: "your_bot_token_here"
  users:
    - "user_id_1"
    - "user_id_2"
  channels:
```

```yaml
      - "@channel_1"
      - "@channel_2"
    adminuser: # 管理员用户配置（可选）
      - "admin_id_1"
      - "admin_id_2"

  rss:
    - urls:
        - "https://example.com/feed1.xml"
        - "https://example.com/feed2.xml"
      interval: 300
      keywords:
        - "keyword1"
        - "keyword2"
      group: "Group1"
      allow_part_match: true # 是否允许部分关键词匹配
```

2.2 配置项说明

2.2.1 配置文件字段说明

配置项	类型	必填	说明	示例
telegram.bot_token	字符串	是	Telegram Bot 的 API Token	"110201543:AAHdqTcvCH1vGWJxfSeofSAs0K5PALDsaw"
telegram.users	字符串数组	是	接收消息的用户 ID 列表	["123456789", "987654321"]
telegram.channels	字符串数组	否	接收消息的频道列表	["@channel1", "@channel2"]
telegram.adminuser	字符串数组	否	管理员用户 ID 列表	["123456789"]
rss[].urls	字符串数组	是	RSS 订阅地址列表	["https://example.com/feed1.xml"]
rss[].interval	整数	是	更新间隔（秒）	300
rss[].keywords	字符串数组	否	关键词列表	["vps", "优惠"]
rss[].group	字符串	否	分组名称	"科技新闻"
rss[].allow_part_match	布尔值	否	是否允许部分匹配	true

2.2.2 环境变量说明

环境变量	必填	说明	示例
TELEGRAM_BOT_TOKEN	是	Telegram Bot 的 API Token	110201543:AAHdqTcvCH1vGWJxfSeofSAs0K5PALDsaw
TELEGRAM_USERS	是	接收消息的用户 ID，多个用逗号分隔	123456789,987654321
TELEGRAM_CHANNELS	否	接收消息的频道，多个用逗号分隔	@channel1,@channel2
TELEGRAM_ADMIN_USERS	否	管理员用户 ID，多个用逗号分隔	123456789,987654321
TELEGRAM_API_URL	否	自定义 Telegram API 服务器地址，用于在无法直接访问官方API的环境中使用代理	http://fyapi.deno.dev/telegram
RSS_URLS	否	RSS 订阅地址，多个组用分号分隔，组内多个地址用逗号分隔。每组可以包含多个 RSS 源，组与组之间用分号隔开。	https://example1.com/feed.xml, https://example2.com/feed.xml; https://example3.com/feed.xml
RSS_KEYWORDS_0	否	第一组 RSS 的关键词，多个用逗号分隔。数字代表组的索引，从 0 开始。	vps,优惠,免费
RSS_INTERVAL_0	否	第一组 RSS 的更新间隔（秒）。数字代表组的索引，从 0 开始。	300
RSS_GROUP_0	否	第一组 RSS 的分组名称。数字代表组的索引，从 0 开始。	科技新闻

环境变量	必填	说明	示例
TZ	否	时区设置，用于确保日志和统计数据使用正确的时区	Asia/Shanghai

2.2.3 配置注意事项

1. **优先级说明**
   - 环境变量的优先级高于配置文件
   - 如果同时设置了环境变量和配置文件，将使用环境变量的值
2. **配置更新机制**
   - 系统每分钟自动检测配置文件变化
   - 配置文件变更后无需重启，自动生效
3. **关键词设置说明**
   - 如果配置了关键词，只有匹配关键词的文章才会被推送
   - 如果没有配置任何关键词，该订阅源的所有新文章都会被推送
   - 关键词匹配支持完整匹配和部分匹配两种模式
   - 可以通过 `allow_part_match` 配置是否允许部分匹配
4. **安全建议**
   - 不要在公开环境中暴露 bot_token
   - 建议设置 adminuser 限制管理权限
   - 定期更新和检查用户权限

2.3 权限说明

系统实现了基本的权限控制机制：

1. 管理员权限：
   - 如果未配置 `adminuser`，所有在 `users` 列表中的用户都具有管理员权限
   - 如果配置了 `adminuser`，则只有在该列表中的用户才具有管理员权限
   - 管理员可以执行所有操作，包括添加/删除用户、管理 RSS 订阅等
2. 普通用户权限：
   - 可以查看所有信息（配置、订阅列表、用户列表等）
   - 不能执行管理操作（添加/删除用户、管理 RSS 订阅）
   - 尝试执行管理操作时会收到提示："您不是系统管理员，无法操作"

2.4 命令说明

机器人支持以下命令：

主要命令：

- `/start` - 开始使用机器人并查看帮助信息
- `/view` - 查看类命令合集
- `/users` - 用户管理命令合集
- `/edit` - 编辑类命令合集

查看类命令（使用 `/view` 查看）：

- `/config` - 查看当前配置
- `/list` - 列出所有 RSS 订阅
- `/stats` - 查看推送统计
- `/version` - 获取当前版本信息

用户管理命令（使用 `/users` 查看）：

- `/add_user` - 添加用户（需要管理员权限）
- `/del_user` - 删除用户（需要管理员权限）
- `/list_users` - 查看用户列表

编辑类命令（使用 `/edit` 查看）：

- `/add` - 添加 RSS 订阅（需要管理员权限）
- `/edit` - 编辑 RSS 订阅（需要管理员权限）
- `/delete` - 删除 RSS 订阅（需要管理员权限）
- `/add_all` - 向所有订阅添加关键词（需要管理员权限）
- `/del_all` - 从所有订阅删除关键词（需要管理员权限）

2.5 环境变量说明

可以通过以下环境变量配置机器人：

```
TELEGRAM_BOT_TOKEN=your_bot_token_here
TELEGRAM_USERS=user_id_1,user_id_2
TELEGRAM_CHANNELS=@channel_1,@channel_2
TELEGRAM_ADMIN_USERS=admin_id_1,admin_id_2  # 管理员用户ID，可选
TELEGRAM_API_URL=http://fyapi.deno.dev/telegram  # 自定义 Telegram API 地址，用于国内代理访问
```

2.6 用户管理

1. 添加用户（需要管理员权限）：

    - 使用 `/add_user` 命令
    - 输入要添加的用户 ID（多个 ID 用空格分隔）
    - 新添加的用户默认为普通用户权限
2. 删除用户（需要管理员权限）：
    - 使用 `/del_user` 命令
    - 查看当前用户列表
    - 输入要删除的用户编号
3. 查看用户列表：
    - 使用 `/list_users` 命令
    - 显示所有已添加的用户 ID
4. 设置管理员：
    - 在配置文件中添加 `adminuser` 字段
    - 或通过环境变量 `TELEGRAM_ADMIN_USERS` 设置
    - 多个管理员 ID 用逗号分隔

注意：如果未设置管理员，所有用户都具有管理员权限。建议在生产环境中明确设置管理员用户。

2.7 Bot 使用方法及命令

Bot 支持以下命令：

- `/start` - 开始使用机器人
- `/help` - 获取帮助信息
- `/config` - 查看当前配置
- `/add` - 添加 RSS 订阅
- `/edit` - 编辑 RSS 订阅
- `/delete` - 删除 RSS 订阅
- `/list` - 列出所有 RSS 订阅
- `/stats` - 查看推送统计

2.8 添加 RSS 订阅

方式一

1. 发送 `/add` 命令给 Bot。
2. 按提示输入 RSS 订阅的 URL。
3. 输入更新间隔（秒）。
4. 输入关键词：
    - 输入 `1`：保持原有关键词（编辑时有效）
    - 输入 `2`：不设置关键词，该订阅源的所有新文章都会被推送

    - 直接输入关键词：输入多个关键词，用空格分隔，只有包含这些关键词的文章才会被推送
5. 输入组名。

方式二

在当前 config 目录下新建 config.ymal，填入以下内容。

```
rss:
  - url: https://rss.nodeseek.com
    interval: 30
    keywords:
      - vps
      - 甲骨文
      - 免费
    group: NS论坛
  - url: https://linux.do/latest.rss
    interval: 30
    keywords:
      - vps
      - 甲骨文
      - 免费
      - 龟壳
    group: LC论坛
```

两种方式都可以，系统会每 1 分钟自动检测，即使动态更改生效。

2.9 编辑 RSS 订阅

1. 发送 `/edit` 命令给 Bot。
2. 输入要编辑的 RSS 订阅编号。
3. 按提示修改 URL、更新间隔、关键词和组名。如果不需要修改某项，直接输入 1。

2.10 删除 RSS 订阅

1. 发送 `/delete` 命令给 Bot。
2. 输入要删除的 RSS 订阅编号。

2.11 查看订阅列表

发送 `/list` 命令给 Bot，查看当前所有 RSS 订阅。

2.12 查看推送统计

发送 `/stats` 命令给 Bot，查看今日和本周的推送数量。

3. 注意事项

- 确保 Docker 容器有足够的权限访问 `config` 和 `data` 目录。

- 如果修改了配置文件，需要重启 Docker 容器以使更改生效。
- 推送统计数据保存在 `/app/data/stats.yaml` 文件中。
- 已发送的项目记录保存在 `/app/data/sent_items.txt` 文件中。

4. 故障排查

- 如果 Bot 无响应，请检查 Telegram Bot Token 是否正确。
- 如果无法接收消息，请确保已将您的用户 ID 添加到配置中。
- 查看 Docker 容器日志以获取更多信息：

```
docker logs rss2tg
```

如有其他问题，请参考项目的 GitHub 页面或提交 issue。

贴上一张效果图

寒山听竹网

这个怎么样帮我看看

https://www.nodeseek.com/post-150954-1

免费

NS论坛

2024-08-24 21:08:44

NodeSeek

这个怎么样帮我看看

https://bill.hostdare.com/store/shared-hosting说的是UnlimitedSSDDiskSpace和UnmeteredBandwidth真的假的，买来做免费...

2 21:09

寒山听竹网

继续158￥出甲骨文新加坡西Gmail原油新号

https://hostloc.com/thread-1338042-1-1.html

甲骨文

HL论坛

2024-08-24 21:11:05

Hostloc

继续158￥出甲骨文新加坡西Gmail原油新号-美国VPS综合讨论-全球主机交流论坛 - Powered by Discuz!

2 21:13

寒山听竹网

[收鸡]BandwagonHost 99刀/年付 The Plan 传奇限量版 VPS

https://www.nodeseek.com/post-150965-1

vps

NS论坛

2024-08-24 21:33:16

NodeSeek

[收鸡]BandwagonHost 99刀/年付 The Plan 传奇限量版 VPS

BandwagonHost2024限量版VPSThePlan传奇限量版VPS

2 21:33

English Version

RSS to Telegram Bot usage documentation

[!TIP] ***Referred to as rss2tg, it is used to instantly send related posts in custom RSS addresses, fields, and refresh times to custom TG users or channels, eliminating the time for you to swipe posts.***

Support AMD64/ARM64

Image size 17M, memory footprint 10M

——By drfyup

---

1. Deployment method

1.1 Use Docker Compose (recommended)

1. Make sure that Docker and Docker Compose are installed (the method is self-searching).
2. Clone or download the project code locally.

```
git clone https://github.com/3377/rss2tg.git
```

3. Enter the project directory.
4. Edit 'docker-compose.yml' file, modify environment variables：

-- Enter any directory or directly the current directory and create a new docker-compose.yml file, fill in the following content

```
version: "3"
services:
  rss2tg:
    container_name: rss2tg
    image: drfyup/rss2tg:latest
    volumes:
      - ./config:/app/config
      - ./data:/app/data
    environment:
      - TELEGRAM_BOT_TOKEN=your_bot_token_here
      - TELEGRAM_USERS=user_id_1,user_id_2
      - TELEGRAM_CHANNELS=@channel_1,@channel_2
      - TZ=Asia/Shanghai
    restart: unless-stopped
```

Replace `your_bot_token_here` with your Telegram Bot Token, `user_id_1, user_id_2` with the user ID you want to receive the message, `@channel_1, @channel_2` with the channel name you want to send the message, `TELEGRAM_API_URL` with your custom Telegram API URL (useful for proxy servers in restricted regions), and `TZ=Asia/Shanghai` with your timezone settings.

5. Run the following command to start the container：

```
docker-compose up  -d
```

1.2 Use Docker Run

1. Build a Docker image：

```
docker pull drfyup/rss2tg:latest
```

2. Run the Docker container：

```
docker run -d \
--name rss2tg \
-v $(pwd)/config:/app/config \
-v $(pwd)/data:/app/data \
-e TELEGRAM_BOT_TOKEN=your_bot_token_here \
-e TELEGRAM_USERS=user_id_1,user_id_2 \
-e TELEGRAM_CHANNELS=@channel_1,@channel_2 \
-e TELEGRAM_API_URL=http://fyapi.deno.dev/telegram \
-e TZ=Asia/Shanghai \
--restart unless-stopped \
drfyup/rss2tg:latest
```

Please replace the corresponding value in the environment variable.

2. Program instructions

2.1 Configuration file

The program supports configuration through YAML configuration files or environment variables.The configuration file is located in`/app/config/config.yaml`.If the file does not exist, the program will use environment variables for initial configuration. The reading priority of environment variables is higher than that of configuration files.

Configuration file example：

```yaml
telegram:
  bot_token: "your_bot_token_here"
  users:
    - "user_id_1"
    - "user_id_2"
  channels:
    - "@channel_1"
    - "@channel_2"
  adminuser: # 管理员用户配置（可选）
    - "admin_id_1"
    - "admin_id_2"

rss:
  - urls:
      - "https://example.com/feed1.xml"
      - "https://example.com/feed2.xml"
    interval: 300
    keywords:
      - "keyword1"
      - "keyword2"
    group: "Group1"
    allow_part_match: true # 是否允许部分关键词匹配
```

2.2 How to use Bot and commands

The Bot supports the following commands：

-`/start`-Start using the robot -`/help`-Get help information -`/config`-View current configuration -`/add`-add RSS subscription -`/edit`-edit RSS feed -`/delete`-delete RSS feed -`/list`-list all RSS feeds -`/statistics`-View push statistics

2.3 Add RSS feed

Method One

1. Send the `/add' command to the Bot.
2. Press the prompt to enter the URL of the RSS subscription.
3. Enter the update interval (seconds).
4. Enter keywords (separated by commas, if not, you can directly enter 1).
5. Enter the group name.

Method Two

Create a new config in the current config directory.ymal, fill in the following.

```yaml
rss:
- url: https://rss.nodeseek.com
  interval: 30
  keywords:
```

```
    - vps
    -Oracle
    -Free
    group: NS Forum
  - url: https://linux.do/latest.rss
    interval: 30
    keywords:
    - vps
    -Oracle
    -Free
    -Turtle shell
    group: LC Forum
```

Both methods are possible, the system will automatically detect every 1 minute, even if the dynamic changes take effect.

2.4 Edit RSS feed

1. Send the `/edit' command to the Bot.
2. Enter the RSS subscription number you want to edit.
3. Follow the prompts to modify the URL, update interval, keywords, and group name.If you don't need to modify an item, enter 1 directly.

2.5 Delete RSS feed

1. Send the `/delete' command to the Bot.
2. Enter the RSS subscription number you want to delete.

2.6 View subscription list

Send the `/list' command to the Bot to view all current RSS feeds.

2.7 View push statistics

Send the `/statistics' command to the Bot to check the number of pushes for today and this week.

3. Precautions

-Make sure that the Docker container has sufficient permissions to access the 'config `and'data` directories. -If the configuration file is modified, the Docker container needs to be restarted for the changes to take effect. -Push statistics are saved in`/app/data/statistics.In the yaml' file. -The sent project records are saved in`/app/data/sent_items.txt` file.

4. Troubleshooting

-If the Bot is unresponsive, please check whether the Telegram Bot token is correct. -If the message cannot be received, please make sure that your user ID has been added to the configuration. -View the Docker container log for more information：

```
docker logs rss2tg
```

If you have other questions, please refer to the project's GitHub page or submit an issue.

Paste a rendering

寒山听竹网

这个怎么样帮我看看

https://www.nodeseek.com/post-150954-1

免费

NS论坛

2024-08-24 21:08:44

NodeSeek

这个怎么样帮我看看

https://bill.hostdare.com/store/shared-hosting说的是UnlimitedSSDDiskSpace和UnmeteredBandwidth真的假的，买来做免费...

2 21:09

寒山听竹网

继续158￥出甲骨文新加坡西Gmail原油新号

https://hostloc.com/thread-1338042-1-1.html

甲骨文

HL论坛

2024-08-24 21:11:05

Hostloc

继续158￥出甲骨文新加坡西Gmail原油新号-美国VPS综合讨论-全球主机交流论坛 - Powered by Discuz!

2 21:13

寒山听竹网

[收鸡]BandwagonHost 99刀/年付 The Plan 传奇限量版 VPS

https://www.nodeseek.com/post-150965-1

vps

NS论坛

2024-08-24 21:33:16

NodeSeek

[收鸡]BandwagonHost 99刀/年付 The Plan 传奇限量版 VPS

BandwagonHost2024限量版VPSThePlan传奇限量版VPS

2 21:33

5. 在中国大陆服务器上使用

在中国大陆服务器上部署 RSS2TG 机器人时，由于网络限制，可能无法直接访问 Telegram 官方 API。为了解决这个问题，本项目支持通过自定义 API URL 使用代理服务进行通信。

5.1 使用代理

1. 设置环境变量 `TELEGRAM_API_URL`，指向可用的代理服务地址，例如：

```
TELEGRAM_API_URL=http://fyapi.deno.dev/telegram
```

2. 在 docker-compose.yml 中配置：

```yaml
environment:
  - TELEGRAM_BOT_TOKEN=your_bot_token_here
  - TELEGRAM_USERS=user_id_1,user_id_2
  - TELEGRAM_API_URL=http://fyapi.deno.dev/telegram
  - TZ=Asia/Shanghai
```

3. 应用重启后，机器人将通过配置的代理地址与 Telegram 进行通信。

5.2 注意事项

- 确保代理服务稳定可靠，否则可能导致消息发送失败
- 定期检查日志，确保通信正常
- 可能需要根据代理服务的要求进行额外设置

如果您使用 `http://fyapi.deno.dev/telegram` 作为代理，通常无需额外配置即可使用。

5. Using in Mainland China

When deploying RSS2TG bot on servers in mainland China, due to network restrictions, direct access to the official Telegram API might be unavailable. To solve this issue, this project supports using a proxy service through a custom API URL.

5.1 Using a Proxy

1. Set the environment variable `TELEGRAM_API_URL` pointing to an available proxy service address, for example:

```
TELEGRAM_API_URL=http://fyapi.deno.dev/telegram
```

2. Configure in docker-compose.yml:

```
environment:
  - TELEGRAM_BOT_TOKEN=your_bot_token_here
  - TELEGRAM_USERS=user_id_1,user_id_2
  - TELEGRAM_API_URL=http://fyapi.deno.dev/telegram
  - TZ=Asia/Shanghai
```

3. After restarting the application, the bot will communicate with Telegram through the configured proxy address.

5.2 Important Notes

- Ensure the proxy service is stable and reliable, otherwise message delivery may fail
- Regularly check logs to ensure communication is normal
- Additional configuration may be required depending on the proxy service's requirements

If you use `http://fyapi.deno.dev/telegram` as a proxy, typically no additional configuration is needed.