RITlug Teleirc Documentation

Release 1.1.2

Mark Repka, Seth Hendrick, Nate Levesque, Robby O'Connor, Jus

Feb 16, 2019

Contents

RITlug Teleirc is a NodeJS implementation of a Telegram <=> IRC bridge. Teleirc works with any IRC channel and Telegram group. It helps bridge communication between a Telegram group and an IRC channel.

This bot was originally written for RITlug and our Telegram group and IRC channel. Today, it is used by various communities other than just RITlug.

CHAPTER 1

Installation

This page is an installation guide for a server administrator to install and run Teleirc. The person who installs Teleirc will have elevated privileges to maintain and administrate the Telegram bot.

There are three parts to installing and configuring Teleirc:

1. Create a Telegram bot
2. Configure and run Teleirc
3. Configure IRC channel

1.1 Create a Telegram bot

Create a new Telegram bot to act as a bridge from the Telegram side. From the bot API, you will receive a token key for the bot. You will then use the bot obtain the unique chat ID of your Telegram group.

1.1.1 Create bot with API

1. Send `/start` in a message to @BotFather[1] user on Telegram
2. Follow instructions from @BotFather to create new bot (e.g. name, username, description, etc.)
3. Receive **token key** for new bot (used to access Telegram API)
4. (*Optional*) Set description or profile picture for your bot with @BotFather
5. (**Required**) Set `/setprivacy` to **DISABLED** (so bot can see messages)[2]
6. Add bot to Telegram group you plan to bridge
7. (*Optional*) Block your bot from being added to more groups (`/setjoingroups`)

[1] @BotFather is the Telegram bot for creating Telegram bots

[2] Privacy setting must be changed for the bot to see messages in the Telegram group. By default, bots cannot see messages unless they use a command to interact with the bot. Since Teleirc forwards all messages, it needs to see all messages. This is why this setting must be changed.

All configuration on Telegram side is complete.

1.2 Configure and run Teleirc

There are several installation options available. This guide details the supported method to configure and run Teleirc. For alternative options, read further below (for Dockerfiles, distro-specific packages, etc.).

1.2.1 Requirements

- git
- nodejs (8.x.x) – we strongly recommend nvm as the installation method.

1.2.2 Download

1. Clone the repository to the server
   - `git clone https://github.com/RITlug/teleirc.git`
2. Change directories to repo
3. Install yarn via one of the following methods (we will only Linux, yarn install documentation cover the rest)
   - Arch Linux

     ```
     pacman -S yarn
     ```

   - Debian/Ubuntu

     ```
     curl -sS https://dl.yarnpkg.com/debian/pubkey.gpg | sudo apt-key add -
     echo "deb https://dl.yarnpkg.com/debian/ stable main" | sudo tee /etc/apt/
     ↪sources.list.d/yarn.list
     sudo apt-get update ; sudo apt-get install --no-install-recommends yarn
     ```

   - Fedora/CentOS/RHEL:

     ```
     curl --silent --location https://dl.yarnpkg.com/rpm/yarn.repo | sudo tee /
     ↪etc/yum.repos.d/yarn.repo
     yum install yarn or dnf install yarn
     ```

4. Install dependencies using yarn
   - `yarn`

1.2.3 Configuration

Teleirc uses dotenv for easy management of API keys and settings. All your configuration changes will live in the `.env` file. You should not need to change other files. This makes it possible to use `git pull` to upgrade the bot in-place.

Explaining config file

The config file you use is a `.env` file. All configuration values for Teleirc are stored here. Copy the example file to a productive file to get started (`cp .env.example .env`) Edit the `.env` file and configure it your preference.

Explanations of the configuration values are below.

IRC

- **IRC_BLACKLIST**:
  - Comma-separated list of IRC nicks to ignore (default: `""`)
- **IRC_BOT_NAME**:
  - Nickname for your bot to use on IRC (default: `teleirc`)
- **IRC_CHANNEL**:
  - IRC channel you want your bot to join (default: `#channel`)
- **IRC_SEND_STICKER_EMOJI**:
  - Send the emoji associated with a sticker on IRC (default: `true`)
- **IRC_PREFIX**:
  - Text displayed before Telegram name in IRC (default: `"<"`)
- **IRC_SUFFIX**:
  - Text displayed after Telegram name in IRC (default: `">"`)
- **IRC_SERVER**:
  - IRC server you wish to connect to (default: `chat.freenode.net`)

Telegram

Note: teleirc **DOES NOT** support channels, only groups. Read more about channels vs groups here.

- **TELEIRC_TOKEN**:
  - Private API token for Telegram bot
- **MAX_MESSAGES_PER_MINUTE**:
  - Maximum rate at which to relay messages (default: `20`)
- **SHOW_ACTION_MESSAGE**:
  - Relay action messages (default: `true`)
- **SHOW_JOIN_MESSAGE**:
  - Relay join messages (default: `false`)
- **SHOW_KICK_MESSAGE**:
  - Relay kick messages (default: `false`)
- **SHOW_LEAVE_MESSAGE**:

  - Relay leave messages (default: `false`)

- **TELEGRAM_CHAT_ID**:
  - Telegram chat ID of the group you are bridging (how do I get this?)

Imgur

- **USE_IMGUR_FOR_IMAGES**:
  - Upload picture messages from Telegram to Imgur, convert picture to Imgur link in IRC (default: `false`)
- **IMGUR_CLIENT_ID**:
  - Imgur API client ID value to access Imgur API

Imgur support

Teleirc retrieves images via the Telegram API. By default, picture messages from Telegram will link to the Telegram API URL. However, the links expire and are not reliable. Optionally, Teleirc can upload an image to Imgur and replace the Telegram API URL with a link to Imgur. This makes picture messages more durable for logs or someone joining the conversation later.

To add Imgur support, follow these steps:

1. Create an Imgur account, if you do not have one
2. Register your bot with the Imgur API
   - Select *OAuth2 without callback* option
3. Put client ID into `.env` file and enable using Imgur

1.2.4 Usage

Choose how you want to run Teleirc persistently. Teleirc officially supports three methods.

pm2

pm2 is a NPM package that keeps NodeJS running in the background. If you run an application and it crashes, pm2 restarts the process. pm2 also restarts processes if the server reboots.

Read the pm2 documentation for more information.

After pm2 is installed, follow these steps to start Teleirc:

```
cd teleirc/
pm2 start -n teleirc-channel teleirc.js
```

systemd

systemd is an option to run the bot persistently. A provided systemd service file is available (`misc/teleirc.service`) Move the provided file to `/usr/lib/systemd/system/` to activate it. Now, you can manage Teleirc through standard `systemctl` commands.

Note that the provided file makes two assumptions:

- Using a dedicated system user (e.g. `teleirc`)
- Home directory located at `/usr/lib/teleirc/` (i.e. files inside Teleirc repository)

screen / tmux

Terminal multiplexers like GNU screen and tmux let you run Teleirc persistently. If you are not familiar with a multiplexer, read more about tmux here.

Inside of your persistent window, follow these steps to start Teleirc:

```
cd teleirc/
node teleirc.js
```

1.2.5 ArchLinux

On ArchLinux, teleirc is available in the AUR. The AUR package uses the systemd method for running Teleirc. Configure the bot as detailed above in the `/usr/lib/teleirc/` directory.

1.2.6 Docker

See also:

See *Using Docker* for more information

1.3 Configure IRC channel

Depending on the IRC network you use, no configuration in IRC is required. However, there are recommendations for best practices to follow.

- Register your channel
- **Give permanent voice to your bridge bot via NickServ (for most networks, the `+Vv` flags)**
  - *Example*: For freenode, `/query NickServ ACCESS #channel ADD my-teleirc-bot +Vv`

CHAPTER 2

Using Docker

Dockerfiles and images are available in `containers/` for configuring and running Teleirc. Install Docker onto the machine you plan to run Teleirc from.

2.1 Which image do I choose?

Official Node Alpine Linux, and Fedora images are provided (ordered ascending by size).

Image	File	Size
Node Alpine Linux (`node:8-alpine`)	`Dockerfile.alpine`	374 MB
Fedora latest	`Dockerfile.fedora`	569 MB

This guide uses `alpine`. If you wish to use `fedora`, replace `alpine` with `fedora`.

You will see errors during `yarn`. You can safely ignore them. They are not fatal.

2.2 Building Docker image

```
cd containers/
docker build . -f Dockerfile.alpine -t teleirc
docker run -d -u teleirc --name teleirc --restart always \
    -e TELEIRC_TOKEN="000000000:AAAAAAaAAa2AaAAaoAAAA-a_aaAAaAaaaAA" \
    -e IRC_CHANNEL="#channel" \
    -e IRC_BOT_NAME="teleirc" \
    -e IRC_BLACKLIST="CowSayBot,AnotherNickToIgnore" \
    -e TELEGRAM_CHAT_ID="-0000000000000" \
    teleirc
```

2.3 Docker Compose

Optionally, you may use docker-compose. We provide an example compose file (`containers/docker-compose.yml`).

```
version: '2'
services:
  teleirc:
    user: teleirc
    build:
      context: ..
      dockerfile: containers/Dockerfile.alpine
    env_file: .env
```

2.3.1 Running with Compose

Run these commands to begin using Teleirc with Docker Compose.

1. Copy `docker-compose.yml.example` to `docker-compose.yml` and edit if you do not wish to use the alpine image
2. Copy `.env.example` to `.env` and edit accordingly.
3. `docker-compose up -d teleirc`

CHAPTER 3

Who uses Teleirc?

The following list of projects and communities use RITlug Teleirc.

- CityZen app (Telegram | IRC)
- **Fedora Project (Telegram | IRC)**
  - Fedora Albania (Telegram | IRC)
  - Fedora CommOps (Telegram | IRC)
  - Fedora Diversity and Inclusion Team (Telegram | IRC)
  - Fedora LATAM (Telegram | IRC)
  - Fedora Marketing Team (Telegram | IRC)
  - Flock to Fedora (Telegram | IRC)
- FOSS@MAGIC at RIT (Telegram | IRC)
- Hamara Linux Project (Telegram | IRC)
- **LibreOffice Community (Telegram | IRC)**
  - LibreLadies (*Telegram invite-only* | IRC)
  - LibreOffice AppImage (*No Telegram @groupname* | IRC)
  - LibreOffice Design Team (*No Telegram @groupname* | IRC)
  - LibreOffice QA Team (*No Telegram @groupname* | IRC)
- MINECON Agents (*Telegram invite-only* | IRC)
- MusicBrainz (Telegram | IRC)
- Open Labs Hackerspace (Telegram | IRC)
- Pure Data (Telegram | IRC)
- **RITlug (*Telegram invite-only* | IRC)**
  - RITlug teleirc (Telegram | IRC)

- Sugar Labs (Telegram | IRC)

3.1 How to get on this list

Want to have your community added to this page? Let us know you're using Teleirc too! Submit an issue against this repo with the following info:

- Organization / group name and website
- Telegram group URL
- Your IRC channel

To be added, your group must not discuss illegal, illicit, or generally inappropriate content.

CHAPTER 4

Live demo

A public Telegram supergroup and IRC channel (on freenode) are available for you to test. Our (small) development community is found in these channels.

- Telegram
- IRC (`#ritlug-teleirc` @ chat.freenode.net)

CHAPTER 5

Indices and tables

- genindex
- modindex
- search