EPFLegram

Nicolas Borboën <nicolas.borboen@epfl.ch>

2015-09-23

EPFLegram

A project to integrate EPFL-API with Telegram, and more. It use Node.js and yagop's telegram bot api.

Telegram

Telegram Messenger is a free and secure messaging for Android and iOS smartphones. You can send messages, photos, videos and documents to people who are in your phone contacts. The Telegram messaging app is cloud-based and cross-platform app which can be accessed via any device. Telegram messages are heavily encrypted and can self-destruct. The apps uses advance encryption technique and decentralized infrastructure which makes the app as one of the fastest messaging apps available in the market.

Initialization

1. Install Telegram
2. Contact @BotFather
3. Create a new bot (/newbot)
4. Generate an authorization token (/token)
5. Check the others options with /help, e.g. /setuserpic, /setcommands, /setabouttext

How Too

Docker way

Pass the authorization token as an environment variable to the docker container:

```
docker run -e "TELEGRAM_BOT_TOKEN=123456789:ABCDEFGHIJKLMOPKRSTUVWXYZ" --name='EPFLegram'  e
```

After an `npm install` the docker will execute the `npm start` and run `node index.js` - the telegram polling is up !

Systemctl

You can run the unit with systemctl (the docker stuff will be done in it)

```
sudo systemctl start EPFLegram.service
```

Fleetctl

Same with fleetctl :

```
fleetctl start EPFLegram.service
```

OldTimer way

`node index.js` or `npm start`

BotFather

- /setcommands
  /menu - Choose between Parmentier, BC, Atlantide, Corbusier or Vinci menus
  /menuAll - All EPFL menus

ToDo

- [] Logs more (i.e. Metro)
- [] Modulize EPFLFunc
- [] Tests

Versions

0.0.3

Refactoring (menu, metro and people are now in seperate file). Generic menu call on behaviors. Thanks to @domq

0.0.2

Integration of @stefanonepa metro api built on transport.opendata.ch

0.0.1

First version with basics commands and tests