i4ins Free TON Notifications Service
tg: @antonsolodkof

Source Code: https://github.com/Strafi/free-ton-notifications-service

Deployed Version: https://ftns.freeton-stats.org/

Demo Server Source Code:
https://github.com/Strafi/free-ton-notifications-service-demo

Deployed Demo Server: https://ftns.freeton-stats.org/test

Wallet:
0:e9fd9b03ae67de6c071510bce975bc5a0b6c5451546e6b941723df5787f95877

i4ins Free TON Notifications Service

Notifications Service for Free TON Blockchain. Service PROD version is already deployed here: https://ftns.freeton-stats.org/ and available via Notifications DeBot (Service ID - i4ins)

Features

- Unique identifier and notification parameters - data is stored in MongoDB
- Guaranteed delivery of notifications - notifications will be delivered even after service shutdown, repeated delivery for 24h in case if webhook consumer unavailable
- Support for HTTPS protocol
- Verification of a domain ownership - via adding a validation file in a .well-known folder
- Logging of events of http notifications for the possibility of displaying them in charts - https://ftns.freeton-stats.org/log/info
- All HTTP API methods must return a 200 response if the requested operation is successful and corresponding HTTP error code otherwise
- Module information - available here https://ftns.freeton-stats.org/info/get
- Well documented with openapi
- http server with UI is provided for testing purposes

I. API Documentation

Api documention and schemas are provided here (openapi): https://ftns.freeton-stats.org/docs/api

II. Testing

Solution has a Demo server for testing purposes

Please, follow the guide in the repository: https://github.com/Strafi/free-ton-notifications-service-demo

III. Installation

1. Clone this repo

```
$ git clone git@github.com:Strafi/free-ton-notifications-service.git i4ins
$ cd i4ins
```

2. Install dependencies

```
$ npm install
```

or

```
$ yarn
```

IV. Environment

To edit environment variables, create a file with name `.env` and copy values below.

Note: if you are going to use docker-compose, you should also update environment in `.yml` files.

Var Name	Type	Default	Description
NODE_ENV	string	`development`	API runtime environment. eg: `staging`
PORT	number	`3000`	Port to run the API server on
MONGO_URL	string	`mongodb://mongo:27017/ftns`	URL for MongoDB
KAFKA_USERNAME	string	`undefined`	Your username
KAFKA_PASSWORD	string	`undefined`	Your password

KAFKA_TOPIC	string	`undefined`	Your topic
KAFKA_GROUP_ID	string	`undefined`	Your group id
KAFKA_BROKER	string	`notification.services.tonlabs.io:29092`	URL for tonlabs kafka broker

V. Run production docker image

1. Add your KAFKA_USERNAME and KAFKA_PASSWORD environment variables

2.1 Using docker-compose

```
$ docker-compose -f docker-compose.prod.yml up
```

2.2 Using docker

```
$ docker pull asolodkov/i4ins
$ docker run -t -i \
      --env NODE_ENV=production \
      --env MONGO_URL=mongodb://mongo:27017/ftns \
      --env KAFKA_USERNAME={YOUR_KAFKA_USERNAME} \
      --env KAFKA_PASSWORD={YOUR_KAFKA_PASSWORD} \
      --env KAFKA_TOPIC={YOUR_KAFKA_TOPIC} \
      --env KAFKA_GROUP_ID={YOUR_KAFKA_GROUP_ID} \
      --env KAFKA_BROKER=notification.services.tonlabs.io:29092 \
      -p 3000:3000 \
      asolodkov/i4ins
```

VI. Build and run from source

The mongo container is only only available in dev environment. When you build and deploy the docker image, be sure to provide the correct environment variables.

1. Build and run without Docker

```
$ npm run build && npm run start
```

or

```
$ yarn build && yarn start
```

2. Run with docker

```
$ docker build -t i4ins .
$ docker run -t -i \
      --env NODE_ENV=production \
      --env MONGO_URL=mongodb://mongo:27017/ftns \
      --env KAFKA_USERNAME={YOUR_KAFKA_USERNAME} \
      --env KAFKA_PASSWORD={YOUR_KAFKA_PASSWORD} \
      --env KAFKA_TOPIC={YOUR_KAFKA_TOPIC} \
      --env KAFKA_GROUP_ID={YOUR_KAFKA_GROUP_ID} \
      --env KAFKA_BROKER=notification.services.tonlabs.io:29092 \
      -p 3000:3000 \
      i4ins
```

3. Run with docker-compose

```
$ docker-compose up
```

VII. Logging

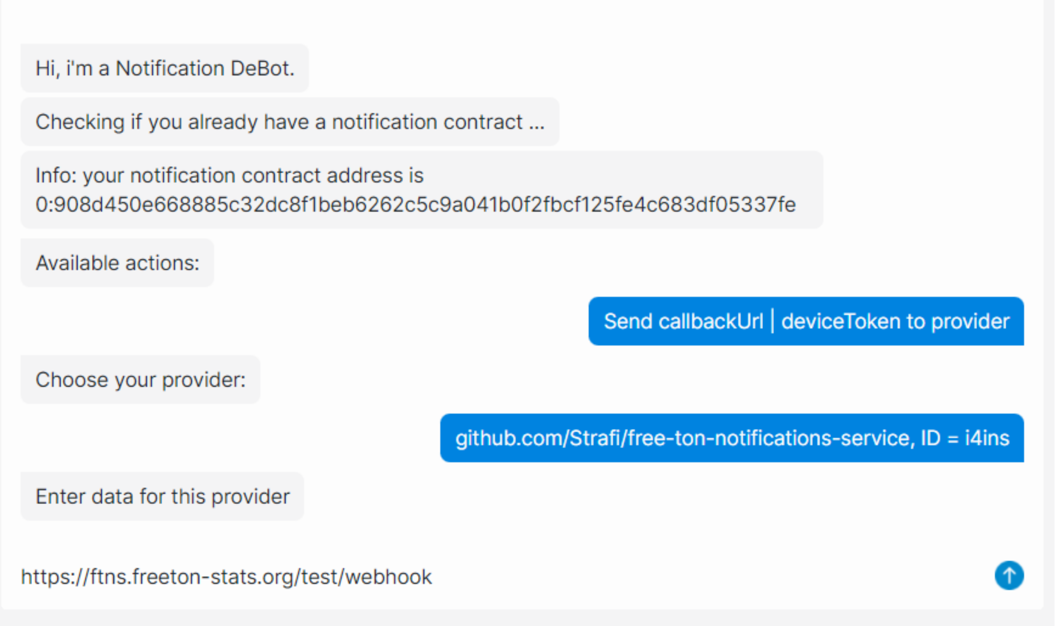
The application uses winston as the default logger.

- All logs are saved in `./logs` directory and at `/logs` in the docker container.
- The `docker-compose` file has a volume attached to container to expose host directory to the container for writing logs.
- Console messages are prettified
- Each line in error log file is a stringified JSON.

Application logs available here: https://ftns.freeton-stats.org/log/info

VIII. Testing

1. Follow the Notification DeBot. User's Manual (dev network) to setup DeBot
2. Send test webhook server url to i4ins provider



Hi, i'm a Notification DeBot.

Checking if you already have a notification contract ...

Info: your notification contract address is
0:908d450e668885c32dc8f1beb6262c5c9a041b0f2fbcf125fe4c683df05337fe

Available actions:

Send callbackUrl | deviceToken to provider

Choose your provider:

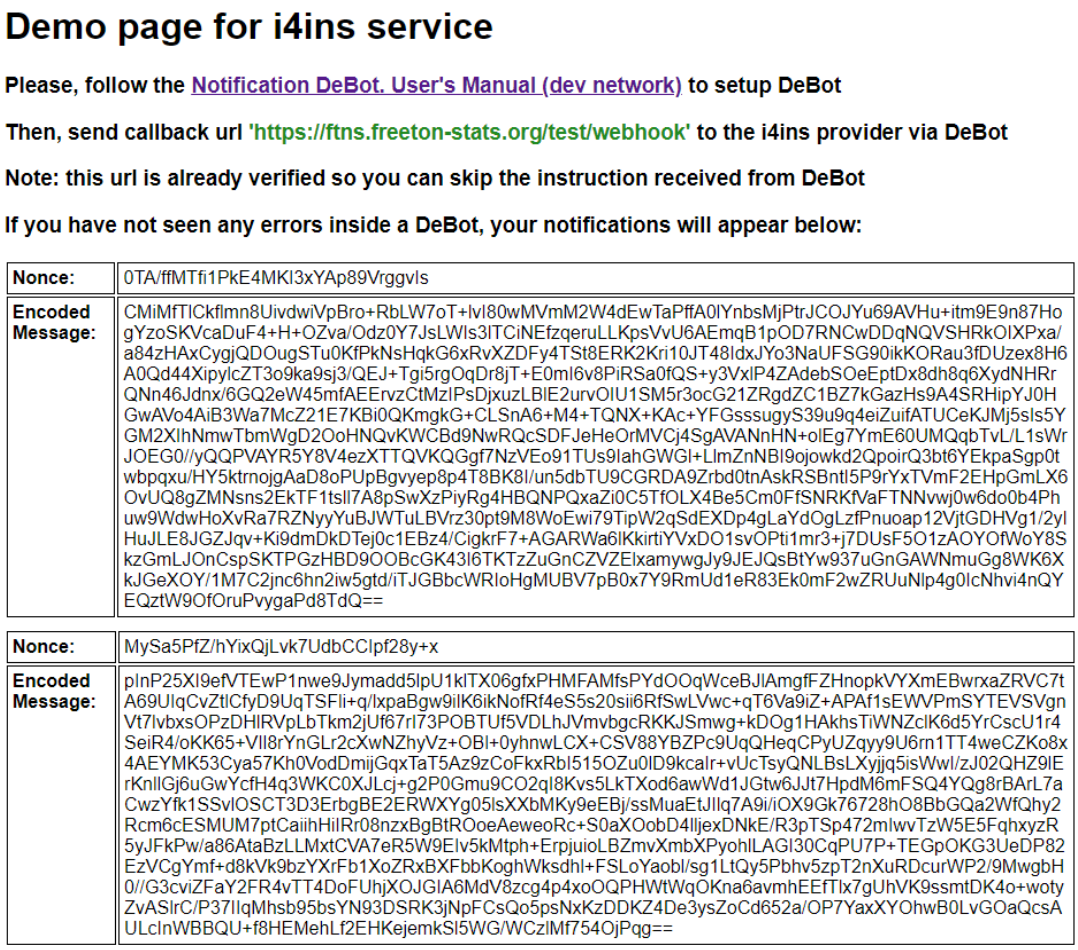
github.com/Strafi/free-ton-notifications-service, ID = i4ins

Enter data for this provider

https://ftns.freeton-stats.org/test/webhook

You will be asked to verify domain ownership, you can skip this step while testing

3. Open https://ftns.freeton-stats.org/test/ . For now you will not see any notifications.
4. Trigger notification according to rules you've set in DeBot
5. You should see a notification on the page



Demo page for i4ins service

Please, follow the Notification DeBot. User's Manual (dev network) to setup DeBot

Then, send callback url 'https://ftns.freeton-stats.org/test/webhook' to the i4ins provider via DeBot

Note: this url is already verified so you can skip the instruction received from DeBot

If you have not seen any errors inside a DeBot, your notifications will appear below:

Nonce:	0TA/ffMTfi1PkE4MKI3xYAp89VrggvIs
Encoded Message:	CMiMfTICkflmn8UivdwiVpBro+RbLW7oT+lvI80wMVmM2W4dEwTaPffA0lYnbsMjPtrJCOJYu69AVHu+itm9E9n87HogYzoSKVcaDuF4+H+OZva/Odz0Y7JsLWls3ITCiNEfzqeruLLKpsVvU6AEmqB1pOD7RNCwDDqNQVSHRkOIXPxa/a84zHAxCygjQDOugSTu0KfPkNsHqkG6xRvXZDFy4TSt8ERK2Kri10JT48IdxJYo3NaUFSG90ikKORau3fDUzex8H6A0Qd44XipylcZT3o9ka9sj3/QEJ+Tgi5rgOqDr8jT+E0mI6v8PiRSa0fQS+y3VxlP4ZAdebSOeEptDx8dh8q6XydNHRrQNn46Jdnx/6GQ2eW45mfAEErvzCtMzIPsDjxuzLBlE2urvOIU1SM5r3ocG21ZRgdZC1BZ7kGazHs9A4SRHipYJ0HGwAVo4AiB3Wa7McZ21E7KBi0QKmgkG+CLSnA6+M4+TQNX+KAc+YFGsssugyS39u9q4eiZuifATUCeKJMj5sIs5YGM2XIhNmwTbmWgD2OoHNQvKWCBd9NwRQcSDFJeHeOrMVCj4SgAVANnHN+oIEg7YmE60UMQqbTvL/L1sWrJOEG0//yQQPVAYR5Y8V4ezXTTQVKQGgf7NzVEo91TUs9IahGWGl+LlmZnNBI9ojowkd2QpoirQ3bt6YEkpaSgp0twbpqxu/HY5ktrnojgAaD8oPUpBgvyep8p4T8BK8I/un5dbTU9CGRDA9Zrbd0tnAskRSBntl5P9rYxTVmF2EHpGmLX6OvUQ8gZMNsns2EkTF1tsIl7A8pSwXzPiyRg4HBQNPQxaZi0C5TfOLX4Be5Cm0FfSNRKfVaFTNNvwj0w6do0b4Phuw9WdwHoXvRa7RZNyyYuBJWTuLBVrz30pt9M8WoEwi79TipW2qSdEXDp4gLaYdOgLzfPnuoap12VjtGDHVg1/2ylHuJLE8JGZJqv+Ki9dmDkDTej0c1EBz4/CigkrF7+AGARWa6lKkirtiYVxDO1svOPti1mr3+j7DUsF5O1zAOYOfWoY8SkzGmLJOnCspSKTPGzHBD9OOBcGK43I6TKTzZuGnCZVZElxamywgJy9JEJQsBtYw937uGnGAWNmuGg8WK6XkJGeXOY/1M7C2jnc6hn2iw5gtd/iTJGBbcWRIoHgMUBV7pB0x7Y9RmUd1eR83Ek0mF2wZRUuNlp4g0IcNhvi4nQYEQztW9OfOruPvygaPd8TdQ==

Nonce:	MySa5PfZ/hYixQjLvk7UdbCCIpf28y+x
Encoded Message:	pInP25XI9efVTEwP1nwe9Jymadd5lpU1kITX06gfxPHMFAMfsPYdOOqWceBJIAmgfFZHnopkVYXmEBwrxaZRVC7tA69UIqCvZtlCfyD9UqTSFli+q/lxpaBgw9iIK6ikNofRf4eS5s20sii6RfSwLVwc+qT6Va9iZ+APAf1sEWVPmSYTEVSVgnVt7lvbxsOPzDHIRVpLbTkm2jUf67rI73POBTUf5VDLhJVmvbgcRKKJSmwg+kDOg1HAkhsTiWNZclK6d5YrCscU1r4SeiR4/oKK65+VlI8rYnGLr2cXwNZhyVz+OBI+0yhnwLCX+CSV88YBZPc9UqQHeqCPyUZqyy9U6rn1TT4weCZKo8x4AEYMK53Cya57Kh0VodDmijGqxTaT5Az9zCoFkxRbI515OZu0lD9kcaIr+vUcTsyQNLBsLXyjjq5isWwl/zJ02QHZ9lErKnlIGj6uGwYcfH4q3WKC0XJLcj+g2P0Gmu9CO2qI8Kvs5LkTXod6awWd1JGtw6JJt7HpdM6mFSQ4YQg8rBArL7aCwzYfk1SSvlOSCT3D3ErbgBE2ERWXYg05lsXXbMKy9eEBj/ssMuaEtJIlq7A9i/iOX9Gk76728hO8BbGQa2WfQhy2Rcm6cESMUM7ptCaiihHiIRr08nzxBgBtROoeAeweoRc+S0aXOobD4IljexDNkE/R3pTSp472mIwvTzW5E5FqhxyzR5yJFkPw/a86AtaBzLLMxtCVA7eR5W9EIv5kMtph+ErpjuioLBZmvXmbXPyohILAGI30CqPU7P+TEGpOKG3UeDP82EzVCgYmf+d8kVk9bzYXrFb1XoZRxBXFbbKoghWksdhl+FSLoYaobl/sg1LtQy5Pbhv5zpT2nXuRDcurWP2/9MwgbH0//G3cviZFaY2FR4vTT4DoFUhjXOJGIA6MdV8zcg4p4xoOQPHWtWqOKna6avmhEEfTlx7gUhVK9ssmtDK4o+wotyZvASlrC/P37IlqMhsb95bsYN93DSRK3jNpFCsQo5psNxKzDDKZ4De3ysZoCd652a/OP7YaxXYOhwB0LvGOaQcsAULclnWBBQU+f8HEMehLf2EHKejemkSl5WG/WCzlMf754OjPqg==