

---

# **FUTURE**

**Roberto Treviño Cervantes**

**Dec 19, 2021**



# CONTENTS

<b>1</b>	<b>Contents</b>	<b>1</b>
1.1	API Endpoints . . . . .	1
<b>2</b>	<b>Quickstart</b>	<b>3</b>



## CONTENTS

### 1.1 API Endpoints

All public FUTURE instances expose a set of dedicated routes that allow to automate the extraction of information. Each one of these simulate as if a user (in this case an application) entered a query through the user interface, but instead reply with JSON.

#### 1.1.1 Output

The most important routes in a FUTURE instance are the **output** endpoints, because they allow to extract information from the server easily. These are used to obtain URL's or images from the index, or perhaps to get a list of all registered peers in a given instance.

##### /\_answer

---

**Note:** The parameters for this route are:

---

This	is	the	main	method	to	get	+	-----	+	-----	+	-----	+	-----	+		Input			Output																		
+	=====	+	=====	+	=====	+	=====	+	=====	+	=====	+	=====	+	=====	+		Name		Type		Name		Type		+	-----	+	-----	+	-----	+	-----	+				
	query		String				+	-----	+	-----	+	-----	+	-----	+		page		Integer				+	-----	+	-----	+	-----	+	-----	+	-----	+					
+	-----	+	-----	+	-----	+	-----	+	-----	+	-----	+	-----	+	-----	+																						

FUTURE is a decentralized, open-source and privacy focused search engine. It is capable of running completely standalone, but it usually complements its own results with others sourced from meta-search at public Searx instances. It also harnesses most of its power when running as a node in a network of independant FUTURE instances, so that they can share and complement their own indexes, thus also providing redundancy to the service. The main instance is located at <https://wearebuildingthefuture.com>



## QUICKSTART

It is easy to setup and run a FUTURE instance publicly so that it contributes to the distributed network. First, you will need to clone the repository:

```
git clone https://github.com/rtrevinnoc/FUTURE.git
cd FUTURE
```

Then you will have to add a `config.py` file, which will allow you to customize important parts of your instance without directly modifying the source code and struggling with updates. It is suggested to start with this configuration template, which is essentially equal to the one used for the main instance:

```
#!/usr/bin/env python3
# -*- coding: utf8 -*-
import secrets
from web3 import Web3
from tranco import Tranco

t = Tranco(cache=True, cache_dir='.tranco')

WTF_CSRF_ENABLED = True
SECRET_KEY = secrets.token_urlsafe(16)
HOST_NAME = "my_public_future_instance" # THE NAMES 'private' and
↳ 'wearebuildingthefuture.com' are reserved for private and main nodes, respectively.
SEED_URLS = ["http://" + x for x in t.list().top(1000)]
PEER_PORT = 3000
HOME_URL = "wearebuildingthefuture.com"
LIMIT_DOMAINS = None
ALLOWED_DOMAINS = []
CONCURRENT_REQUESTS = 10
CONCURRENT_REQUESTS_PER_DOMAIN = 2.0
CONCURRENT_ITEMS = 100
REACTOR_THREADPOOL_MAXSIZE = 20
DOWNLOAD_MAXSIZE = 10000000
AUTOTHROTTLLE = True
TARGET_CONCURRENCY = 2.0
MAX_DELAY = 30.0
START_DELAY = 1.0
DEPTH_PRIORITY = 1
LOG_LEVEL = 'INFO'
CONTACT = "rtrevinnoc@wearebuildingthefuture.com"
MAINTAINER = "Roberto Treviño Cervantes"
FIRST_NOTICE = "Written and Mantained By <a href='https://keybase.io/rtrevinnoc'>Roberto_
↳ Treviño</a>"
```

(continues on next page)

(continued from previous page)

```
SECOND_NOTICE = "Proudly Hosted on <a href='https://uberspace.de/en/'>Uberspace</a>"
DONATE = "<a href='https://www.buymeacoffee.com/searchatfuture'>DONATE</a>"
COLABORATE = "<a href='https://github.com/rtrevinnoc/FUTURE'>COLABORATE</a>"
CACHE_TIMEOUT = 15
CACHE_THRESHOLD = 100
COMPLEMENTARY_VECTOR_CACHE = -1
try:
    WEB3API = Web3(Web3.HTTPProvider('http://127.0.0.1:8545'))
    ETHEREUM_ACCOUNT = WEB3API.eth.accounts[0]
    CONTRACT_CODE = 'future-token/build/contracts/FUTURE.json'
    CONTRACT_ADDRESS = "0x2ebDA3D6B2F24aE57164b0384daa9af2C0D17323"
except:
    pass
```

**NOTE:** In case you want to use a docker container, simply run the following commands before everything else below:

```
docker build -t future .
docker run -i -t -p 3000:3000 future bash
```

After you have configured your FUTURE instance, but before you can start the server, you will be required to add a minimum of ~25 urls to your local index, by executing:

```
chmod +x bootstrap.sh
./bootstrap.sh
./build_index.sh
```

At any point in time, you can check how much webpages are in your local index by executing:

```
python3 count_index.py
```

And eventually, you can interrupt the crawler by executing:

```
./save_index.sh
```

Naturally, you can restart it using `./build_index.sh`. And with this, you can start your development server with:

```
./future.py
```

However, if you are planning to contribute to the shared index by making your instance public, it is recommended to use uWSGI. We suggest using this configuration template, with `touch uwsgi.ini`, as it is used on the main instance.

```
[uwsgi]
module = future:app
pidfile = future.pid
http-socket = :3000
chmod-socket = 660
strict = true
master = true
enable-threads = true
vacuum = true ; Delete sockets during shutdown
single-interpreter = true
die-on-term = true ; Shutdown when receiving SIGTERM (default is res
↳respawn)
```

(continues on next page)



(continued from previous page)

```

need-app = true

disable-logging = true          ; Disable built-in logging
log-4xx = true                 ; but log 4xx's anyway
log-5xx = true                 ; and 5xx's

cheaper-algo = busyness
processes = 6                  ; Maximum number of workers allowed
cheaper = 1                    ; Minimum number of workers allowed
cheaper-initial = 2            ; Workers created at startup
cheaper-overload = 1           ; Length of a cycle in seconds
cheaper-step = 1               ; How many workers to spawn at a time

cheaper-busyness-multiplier = 30 ; How many cycles to wait before killing workers
cheaper-busyness-min = 20        ; Below this threshold, kill workers (if stable for ↵
↵multiplier cycles)
cheaper-busyness-max = 70        ; Above this threshold, spawn new workers
cheaper-busyness-backlog-alert = 4 ; Spawn emergency workers if more than this many ↵
↵requests are waiting in the queue
cheaper-busyness-backlog-step = 2 ; How many emergency workers to create if there are ↵
↵too many requests in the queue

```

Finally, start your public node to contribute to the shared network with the following command:

```
uwsgi uwsgi.ini
```