

---

# **espyta Documentation**

***Release 0.0.0***

**supakeen**

**Nov 01, 2020**



# CONTENTS

<b>1</b>	<b>Introduction</b>	<b>3</b>
<b>2</b>	<b>Usage</b>	<b>5</b>
<b>3</b>	<b>Installation</b>	<b>7</b>
<b>4</b>	<b>Contributing</b>	<b>9</b>
<b>5</b>	<b>Table of Contents</b>	<b>11</b>
5.1	Installation . . . . .	11
5.2	Configuration . . . . .	13
5.3	Autodoc . . . . .	13
	<b>Python Module Index</b>	<b>15</b>
	<b>Index</b>	<b>17</b>



Welcome to the documentation of the `espyta` ESP OTA webhook software.



## INTRODUCTION

`espyta` is a straightforward web application written for Python 3.6 and up using the [Tornado](#) web framework.





## USAGE

You can run the built in `espyta` HTTP server through the `espyta http` command. This will serve up a HTTP server listening on localhost on port 8000.



## INSTALLATION

You can install `espyta` from [PyPI](#) by running `pip` as follows:

```
pip install espyta
```

I suggest you use a virtual environment for installation. There are extended [Installation](#) instructions available which explain how to do so.



## **CONTRIBUTING**

`espyta` is a place that will accept your first contribution to an open source project. The preferred place to start out is to visit our [GitHub](#) page and look at the [issues](#) there. If you can solve any of them then you can send a pull request. I will make sure to review your code.

If you are thinking about contributing a new feature then keep in mind that `espyta` is trying to stay as small and lean of a project as possible. Open a ticket first if you have a specific feature in mind.



## TABLE OF CONTENTS

### 5.1 Installation

The common way of installing `espyta` is by installing from PyPI. I suggest you to use a virtual environment, these prevent accidentally updating libraries that your other projects or even your operating system depend on.

For a Debian based distribution installation would look like this:

```
python3 -m venv virtual-environment virtual-environment/bin/pip install espyta
```

After this you can run `espyta` in the following way:

```
virtual-environment/bin/espyta http
```

This will start the built-in HTTP server on localhost, port 8000.

#### 5.1.1 Running on boot

If you wish to run `espyta` as an actual service there's a few more things we will need to take care of. This bit of the documentation has strong ideas about how to host a website and doesn't provide commands but general guidance.

This guide only applies to Linux and BSD systems though I am sure it can be applied to Windows systems as well if you read synonymous terms where necessary.

#### Prerequisites

- A HTTP server or proxy such as nginx, haproxy, apache2, etc.

#### Setup

Start by creating a separate user for your `espyta` this way we can make use of the filesystem.

After you've created this user it's now time to setup an environment where this users code will live. Let's get into their homedir and then perform the initial section of this document as the user you've just created.

From now on I'll assume you have `/home/youruser/virtual-environment` with `espyta` installed to it.

We'll create a configuration file `/home/youruser/espyta.toml` with the following content:

```
database_uri = "mysql+pymysql://user:password@host/database"
```

If you want to configure more then read the [Configuration](#) section.

Now that we have all of this setup it's time to test out `espyta` real quick:

```
/home/youruser/virtual-environment/bin/espyta --configuration-path /home/youruser/
↳ espyta.toml http
```

This should start espyta listening on localhost port 8000. Verify that this is the case.

I will use systemd for this example since it comes pre-installed on most of our systems nowadays.

Take the example systemd unit file from the [repository](#) and place it in `/etc/systemd/system/espyta.service`. Then open the file and adjust the paths to the paths you've created.

After you've done this you can `systemctl daemon-reload` and `systemctl enable espyta.service`. Check its status, if it has come up verify that you can connect to localhost port 8000 as well and get served with the espyta homepage.

Now it's time to configure our webserver to forward requests to espyta. I'll use nginx in this example but the ideas carry over to anything you might be using.

Here's an example nginx configuration file:

```
server {
    listen 443 ssl ;
    listen [::]:443 ssl ;

    root /var/www/empty;

    server_name myespyta.net; # managed by Certbot

    add_header X-Xss-Protection "1; mode=block" always;
    add_header X-Content-Type-Options "nosniff" always;
    add_header X-Frame-Options "SAMEORIGIN" always;
    add_header Content-Security-Policy "default-src 'self'" always;
    add_header Strict-Transport-Security "max-age=31536000; includeSubdomains;
↳ preload" always;
    add_header Referrer-Policy "no-referrer" always;
    add_header Feature-Policy "accelerometer 'none'; camera 'none'; geolocation
↳ 'none'; gyroscope 'none'; magnetometer 'none'; microphone 'none'; payment 'none';
↳ usb 'none'" always;

    location / {
        limit_req zone=myespyta burst=100;
        proxy_pass http://127.0.0.1:8000;
        proxy_set_header Host $host;
        proxy_set_header X-Forwarded-Proto https;
    }

    access_log /home/youruser/myespyta.net_access.log;

    ssl_certificate /etc/letsencrypt/live/myespyta.net/fullchain.pem; # managed
↳ by Certbot
    ssl_certificate_key /etc/letsencrypt/live/myespyta.net/privkey.pem; # managed
↳ by Certbot
}
```

Place that file in `/etc/nginx/sites-enabled/myespyta.net` and reload your nginx. It is important that you pass the Host header and protocol as espyta will use these to build its URLs.



## 5.2 Configuration

espyta is configured in two ways, one is by arguments and the other is through a configuration file.

The options available are dependent on the command you're running. You can always pass the `--configuration-path` argument to espyta.

Here is a quick example:

```
espyta --configuration-path /tmp/foo.toml http --port 9000
```

The `http` subcommand takes a separate argument `--port` to override the default listening port (8000).

### 5.2.1 File

The configuration file has a bunch more properties to configure espyta with. Here's an example file.

### 5.2.2 Options

## 5.3 Autodoc

### 5.3.1 espyta.command

### 5.3.2 espyta.http

### 5.3.3 espyta.configuration



## PYTHON MODULE INDEX

### e

`espyta.configuration`, [13](#)



## INDEX

### E

`espyta.configuration`  
module, [13](#)

### M

module  
  `espyta.configuration`, [13](#)