# LEARNING

# tornado

#tornado

# Table of Contents

# About

You can share this PDF with anyone you feel could benefit from it, downloaded the latest version from: tornado

It is an unofficial and free tornado ebook created for educational purposes. All the content is extracted from Stack Overflow Documentation, which is written by many hardworking individuals at Stack Overflow. It is neither affiliated with Stack Overflow nor official tornado.

The content is released under Creative Commons BY-SA, and the list of contributors to each chapter are provided in the credits section at the end of this book. Images may be copyright of their respective owners unless otherwise specified. All trademarks and registered trademarks are the property of their respective company owners.

Use the content presented in this book at your own risk; it is not guaranteed to be correct nor accurate, please send your feedback and corrections to info@zzzprojects.com

# Chapter 1: Getting started with tornado

## Remarks

Tornado is a Python web framework and asynchronous networking library, that uses non-blocking network I/O which allows it to scale to tens of thousands of open connections, making it ideal for long polling, WebSockets, and other applications that require a long-lived connection to each user.

## Versions

Release Notes

## Examples

### Installation or Setup

Python3 - `sudo pip3 install tornado`
Python2 - `sudo pip install tornado`

Packages which will are optional but recommended to install alongside Tornado :

- concurrent.futures
- pycurl
- pycares
- Twisted
- monotonic or monotime

### Hello World

```
# hello_server.py
import tornado.ioloop
import tornado.web

class MainHandler(tornado.web.RequestHandler):
    def get(self):
        self.write("Hello, world")

    def make_app():
        return tornado.web.Application([ (r"/", MainHandler), ])  # URL Mapping

if __name__ == "__main__":
    app = make_app()
    app.listen(8888)    # Port Number
    tornado.ioloop.IOLoop.current().start()
```

This App is run by typing `python3 hello_server.py` or `python hello_server.py` depending on the version of Python being used.

When run locally the server can be accessed by going to `127.0.0.1:8888` from the browser.
The server will return "Hello World".
In `make_app()` function, the root `/` is mapped to `MainHandler`. This means that requests to the root IP `127.0.0.1:8888` will be mapped to the `MainHandler` function.

Read Getting started with tornado online: https://riptutorial.com/tornado/topic/5269/getting-started-with-tornado

# Credits

| S. No | Chapters | Contributors |
|---|---|---|
| 1 | Getting started with tornado | Community, Meghdeep Ray |