



**Tornado**

The logo for Friendfeed, featuring the word "friendfeed" in a blue, rounded, lowercase font with a slight shadow effect.The Facebook logo, consisting of the word "facebook" in white lowercase letters on a dark blue rectangular background.

- Tornado, Friendfeed'in kullandığı non-blocking (bkz:non blocking) ve ölçeklenebilir web araçlarının adıdır.
- Facebook, Friendfeed'i satın aldıktan Apache lisansı ile açık kaynak olarak dağıtılmaya başladı.



# Kurulum

```
wget -c http://github.com/downloads/facebook/tornado/tornado-1.0.tar.gz
```

```
tar xvzf tornado-1.0.tar.gz
```

```
cd tornado-1.0
```

```
python setup.py build
```

```
sudo python setup.py install
```

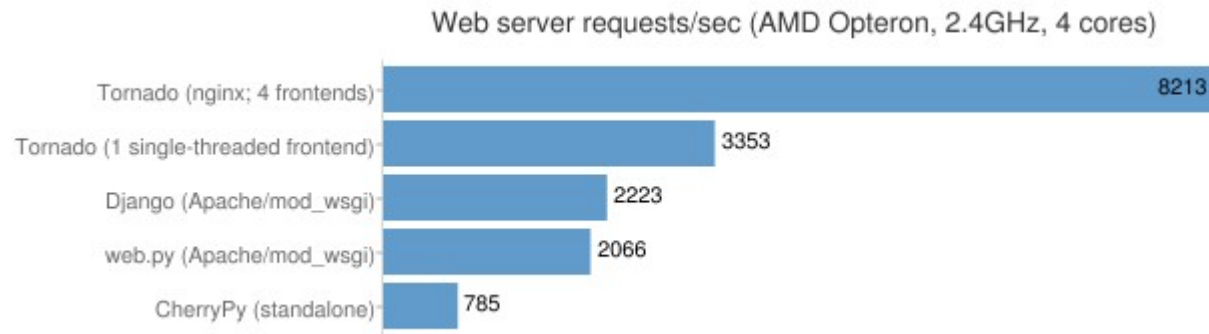
Veya

```
apt-get install python-tornado (debian sid)
```



# Performans

- 'Hello world' performansi





# Modüller

## High Level Modüller

- web
- escape
- database
- template
- httpclient
- auth
- locale
- options

## Low level modüller

- httpserver
- iostream
- ioloop
  
- epoll



web

Friendfeed'in üzerine kurulduđu, başlıca ana parçaları içeren modül



escape

XHTML, JSON ve URL encode/decode işlemleri için modüller



MySQLdb modülü üzerine geliştirilmiş bir wrapper



Şablon motoru



| auth

### 3. parti kimlik doğrulama kütüphaneleri

(Google OpenID/OAuth, Facebook Platform, Yahoo BBAuth, FriendFeed OpenID/OAuth, Twitter OAuth)



locale

Yerelleştirme desteği için modül



options

Komut satırından veya yapılandırma dosyasından yapılandırma almak için modül (optparser)



low level

Bunların altında ise esas non-blocking işlemleri yapan

- httpserver : non-blocking http sunucu
  - iostream : non-blocking socketleri okumak için geliştirilmiş bir modül
  - ioloop : ana i/o modülü
- 
- Bir de amazon s3 simüle eden bir sunucu modülü var.



kod örnekleri

<http://github.com/yuxel/snippets/tree/master/python/tornado/>



hello world

```
#!/usr/bin/python

import tornado.httpserver
import tornado.ioloop
import tornado.web

class HelloHandler(tornado.web.RequestHandler):
    def get(self):
        self.write("Hello World")

application = tornado.web.Application([
    (r"/", HelloHandler),
])

if __name__ == "__main__":
    http_server = tornado.httpserver.HTTPServer(application)
    http_server.listen(8888)
    tornado.ioloop.IOLoop.instance().start()
```

<http://github.com/yuxel/snippets/blob/master/python/tornado/helloWorld.py>



# request handling

```
.....  
  
# handles pages matches URI /another/someUriParam  
class AnotherHandler(tornado.web.RequestHandler):  
    def get(self,uriParam):  
        self.write("This is another page with URI parameter = " + uriParam)  
  
application = tornado.web.Application([  
    (r"/another/([^/]+)", AnotherHandler),  
])  
  
.....
```

<http://github.com/yuxel/snippets/blob/master/python/tornado/requestHandling.py>



# request handling

```
...
# handles pages matches /postTest/ and demonstrate getting request parameter
class PostSomethingHandler(tornado.web.RequestHandler):
    def get(self):
        form = """<form method="post">
        <input type="text" name="something"/>
        <input type="submit"/>
        </form>"""
        self.write(form)

    def post(self):
        postedValue = self.get_argument("something") # get request argument
        self.write("You've postted 'something' as : " + postedValue)

application = tornado.web.Application([
    (r"/postTest/", PostSomethingHandler),
])
...
```

<http://github.com/yuxel/snippets/blob/master/python/tornado/requestHandling.py>



# request handling

```
...
# demonstrates an HTTP response error
class ResponseErrorHandler(tornado.web.RequestHandler):
    def get(self):
        # send a 403 forbidden error
        raise tornado.web.HTTPError(403)

application = tornado.web.Application([
    (r"/someError/", ResponseErrorHandler),
])
```

...

<http://github.com/yuxel/snippets/blob/master/python/tornado/requestHandling.py>



# cookie

...

# demonstrates cookie usage

```
class MainHandler(tornado.web.RequestHandler):
    def get(self):
        cookieName = "pyist";
        cookieValue = self.get_cookie(cookieName)
        currentTimestamp = str(time.time())
        if not cookieValue:
            self.set_cookie(cookieName, currentTimestamp)
            self.write("I've just set your cookie, refresh!")
        else:
            self.write("Cookie value : " + cookieValue)
```

...

<http://github.com/yuxel/snippets/blob/master/python/tornado/cookie.py>



# secure cookie

...

# demonstrates secure cookie example

```
class SecureCookieHandler(tornado.web.RequestHandler):
```

```
    def get(self):
```

```
        cookieName = "pyist_secure";
```

```
        cookieValue = self.get_secure_cookie(cookieName)
```

```
        currentTimestamp = str(time.time())
```

```
        if not cookieValue:
```

```
            self.set_secure_cookie(cookieName, currentTimestamp)
```

```
            self.write("I've just set your cookie, refresh!")
```

```
        else:
```

```
            self.write("Cookie value : " + cookieValue)
```

...

<http://github.com/yuxel/snippets/blob/master/python/tornado/cookie.py>



# template

```
...
<!DOCTYPE html>
<html>
  <head>
    <title>{{ title }}</title>
    <meta charset="utf-8" />
  </head>
  <body>
    {% if userLogged %}
      Hi {{ userLogged }}
    {% else %}
      You need to log in to see this page
    {% end %}
  </body>
</html>
```

...

<http://github.com/yuxel/snippets/blob/master/python/tornado/templates/main.html>



# template

...

# handles main page

```
class MainHandler(tornado.web.RequestHandler):
```

```
    def get(self):
```

```
        #we demonstrate this using a ?userLogged=userName parameter
```

```
        switchLoggedFromGet = self.get_argument("userLogged", False)
```

```
        #remove html entities
```

```
        switchLoggedFromGet = tornado.escape.xhtml_escape(switchLoggedFromGet)
```

```
        self.render("templates/main.html", title="Pyist.net", userLogged=switchLoggedFromGet)
```

...

<http://github.com/yuxel/snippets/blob/master/python/tornado/template.py>



# locale

```
...
class TRHandler(tornado.web.RequestHandler):
    def get(self):
        tornado.locale.set_default_locale('tr_TR')
        self.render("templates/translation.html")

# English page
class ENHandler(tornado.web.RequestHandler):
    def get(self):
        tornado.locale.set_default_locale('en_US')
        self.render("templates/translation.html")

...
if __name__ == "__main__":
    #set path for location dir
    translationsPath = os.path.join(os.path.dirname(__file__), "translations")
    tornado.locale.load_translations(translationsPath)

...
```

<http://github.com/yuxel/snippets/blob/master/python/tornado/translation.py>



locale

tr\_TR.csv

"Hello World!","Merhaba dunya!"

"This is another test","Bu bir test"

en\_EN.csv

"Hello World!","Hello, world!"

"This is another test","This is another test"



# xsrif koruması

```
...
# handles main page
class MainHandler(tornado.web.RequestHandler):
    def get(self):
        self.render("templates/xsrf.html")

# if _xsrf value doesnt match xsrf cookie value, this will return 403
def post(self):
    postedValue = self.get_argument("something") # get request argument
    self.write("You've postted 'something' as : " + postedValue)

application = tornado.web.Application([
    (r"/", MainHandler),
], cookie_secret="SomeSecret",xsrf_cookies=True)
...
```

<http://github.com/yuxel/snippets/blob/master/python/tornado/xsrf.py>



# xsrif koruması

```
<!DOCTYPE html>
<html>
  <head>
    <title>Pyist Tornadoweb</title>
    <meta charset="utf-8" />
  </head>
  <body>
    <form action="" method="post">
      {{ xsrf_form_html() }}
      <div>
        Write something: <input type="text" name="something"/>
        <input type="submit" value="Try"/>
      </div>
    </form>
  </body>
</html>
```

<http://github.com/yuxel/snippets/blob/master/python/tornado/templates/xsrf.html>



# asenkron istekler

```
import tornado.httpclient
...
class MainHandler(tornado.web.RequestHandler):
    @tornado.web.asynchronous
    def get(self):
        http = tornado.httpclient.AsyncHTTPClient()
        url = "http://api.eksigator.com/demo/demo/getList"
        http.fetch(url, callback=self.async_callback(self.on_response))

    def on_response(self, response):
        items = tornado.escape.json_decode(response.body)
        for item in items:
            self.write( item["title"] + "<br/>" )

        # this will end async request
        self.finish()
...
```

<http://github.com/yuxel/snippets/blob/master/python/tornado/async.py>



# database

```
import tornado.database
```

```
...
```

```
class DBHandler(tornado.web.RequestHandler):
```

```
    def get(self):
```

```
        db = tornado.database.Connection(
            host="localhost", database="pyist",
            user="root", password="12345678")
```

```
        datas = db.query("select name_surname from pyist")
```

```
        # print datas
```

```
        for data in datas:
```

```
            self.write(data["name_surname"])
            self.write("<br/>")
```

```
...
```

<http://github.com/yuxel/snippets/blob/master/python/tornado/database.py>



# static dosyalar

...

```
application = tornado.web.Application([  
    (r"/", MainHandler),  
], static_path = os.path.join(os.path.dirname(__file__), "static"))
```

...

```
{{ static_url("hello.txt") }}
```

<http://github.com/yuxel/snippets/blob/master/python/tornado/translation.py>



# Tornado

# chat demo

## Chat demosunu sadeleştirilmiş hali



pyist.net

Send!

10 ağustos  
pyist  
cevahir  
toplantı

<http://github.com/yuxel/snippets/blob/master/python/tornado/chat/>



<http://github.com/yuxel/feedget>

<http://feedget.net/> (alpha)



**Tornado**

kaynaklar

<http://www.tornadoweb.org/>

<http://github.com/facebook/tornado>

<irc://irc.freenode.net/tornado>