

Tornado

- Web Frameworks
- Basic Handler
- Static Content
- GET, POST, and more

Advanced Python Programming



See Also

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

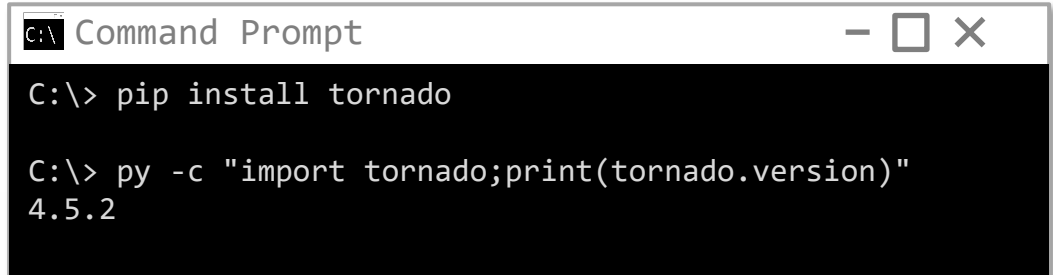
https://www.youtube.com/watch?v=DwrYo_YU9wM



Tornado

Python Web Frameworks

- Hide the low-level socket/thread details
- Some of the popular frameworks:
 - Django
 - TurboGears
 - web2py
 - Falcon
 - Flask
- We will use Tornado



```
C:\> pip install tornado

C:\> py -c "import tornado;print(tornado.version)"
4.5.2
```

Basic Tornado Server

```
import tornado.ioloop
import tornado.web
```

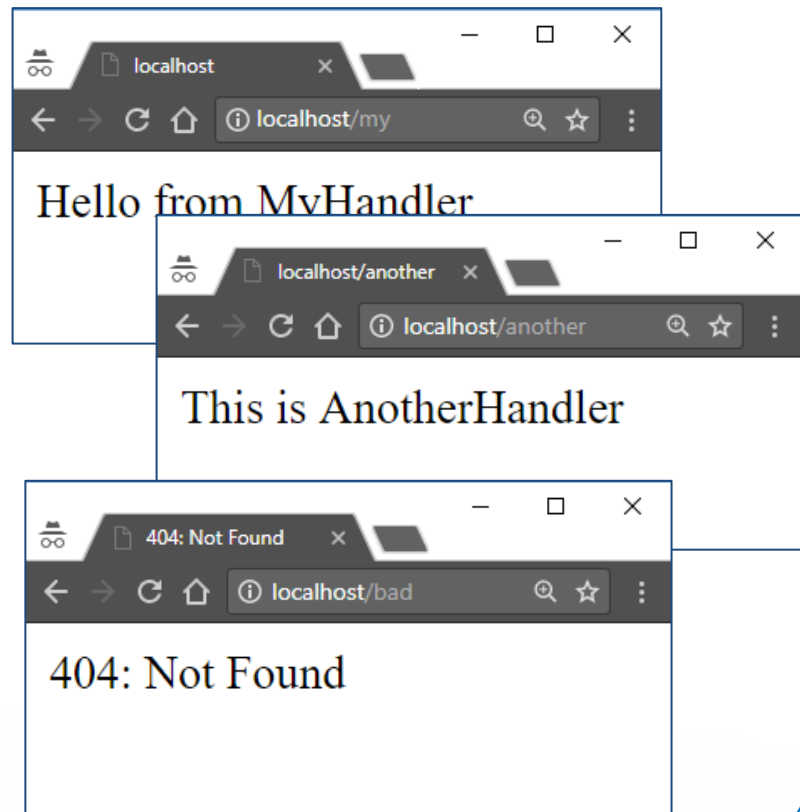
```
class MyHandler(tornado.web.RequestHandler):
    def get(self):
        self.write("Hello from MyHandler")
```

```
class AnotherHandler(tornado.web.RequestHandler):
    def get(self):
        self.write("This is AnotherHandler")
```

```
handlers = [
    ("/my", MyHandler),
    ("/another", AnotherHandler)
]
```

List of Tuples

```
app = tornado.web.Application(handlers)
app.listen(80)
tornado.ioloop.IOLoop.current().start()
```

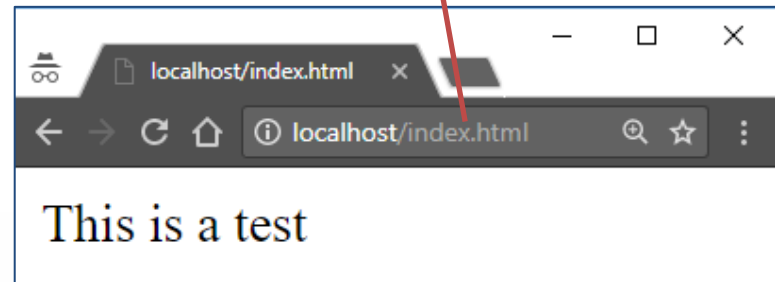
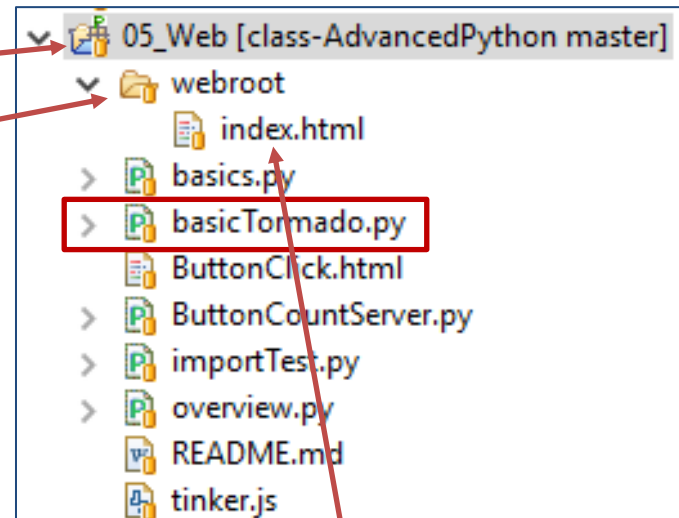


Static Content

```
# The directory of this module
root = os.path.dirname(__file__)
root = os.path.join(os.path.dirname(__file__), "webroot")

handlers = [
    ("/my", MyHandler),
    ("/another", AnotherHandler),
    (r"/(.*)", tornado.web.StaticFileHandler, {"path": root}),
]

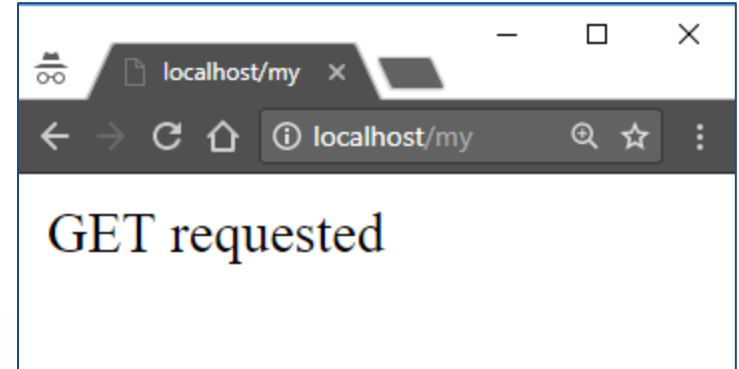
app = tornado.web.Application(handlers)
app.listen(80)
tornado.ioloop.IOLoop.current().start()
```



HTTP Verbs and Postman

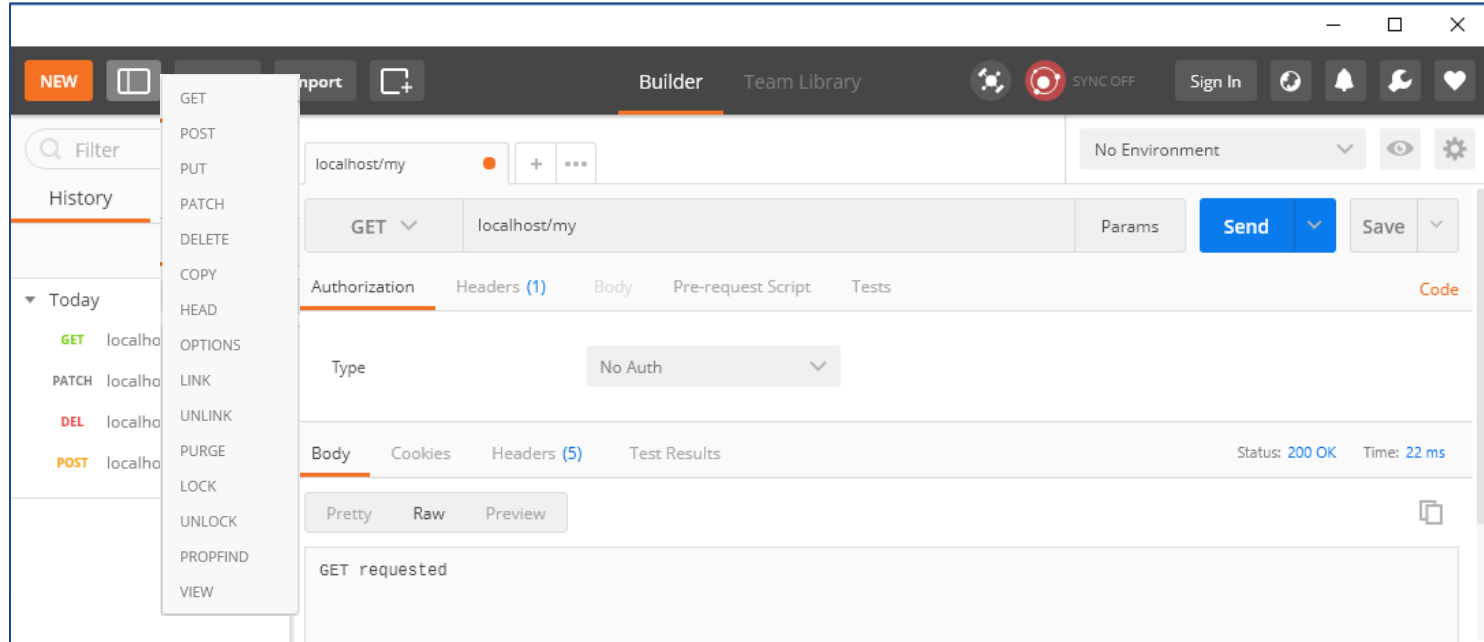
- Traditional browsing sends GETs and POSTs (forms)
- Other verbs are used in specialized applications (like REST)
- You can code to handle one or more verbs

```
class MyHandler(tornado.web.RequestHandler):  
    def get(self):  
        self.write("GET requested")  
  
    def post(self):  
        self.write("POST requested")  
  
    def delete(self):  
        self.write("DELETE requested")
```



Postman

- “Postman” is a Chrome application
- Great for testing/debugging HTTP transactions



Postman

History Collections

Clear all

Today

- POST localhost/my
- PATCH localhost/my
- POST localhost/my
- GET localhost/my
- PATCH localhost/my
- DEL localhost/my
- POST localhost/my

POST localhost/my

Authorization Headers (1) Body

Type No

Body Cookies Headers (4)

Pretty Raw Preview HTML

i 1 POST requested

PATCH localhost/my

Body Cookies Headers (4) Test Results

Pretty Raw Preview HTML

```
i 1 <html>
2   <title>405: Method Not Allowed</title>
3   <body>405: Method Not Allowed</body>
4 </html>
```


HTTP Headers

```
class MyHandler(tornado.web.RequestHandler):  
    def get(self):  
        print(self.request)  
        print(str(self.request.headers))  
        print(self.request.headers["Accept-Language"])  
        self.write("GET requested")
```

```
HTTPServerRequest(protocol='http', host='localhost', method='GET', uri='/my',  
version='HTTP/1.1', remote_ip='::1', headers={ ... })
```

Accept-Language: en-US,en;q=0.8

Accept-Encoding: gzip, deflate, br

Connection: keep-alive

Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,*/*;q=0.8

User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko)

Chrome/61.0.3163.100 Safari/537.36

Host: localhost

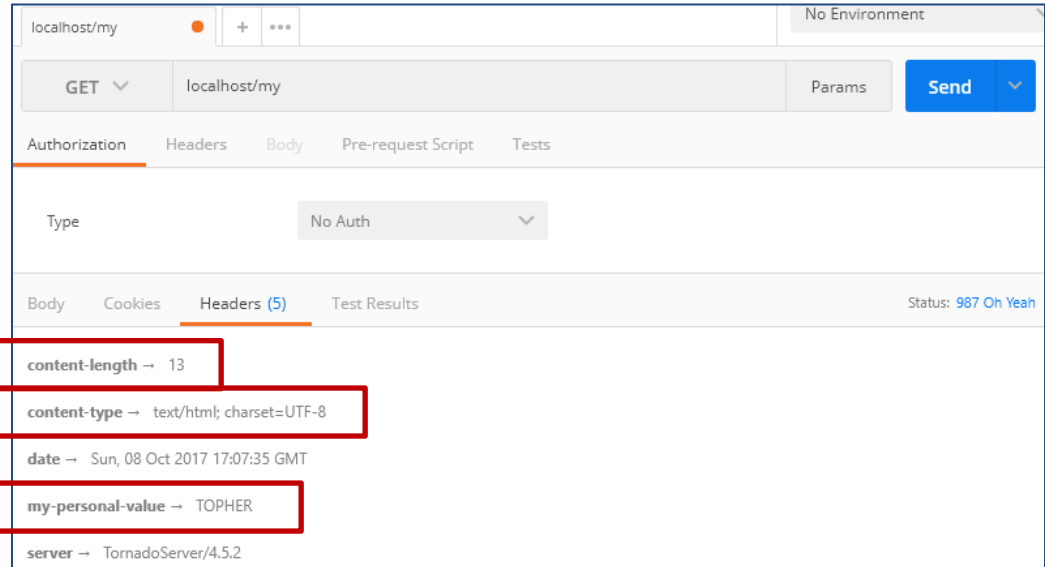
Cache-Control: max-age=0

Upgrade-Insecure-Requests: 1

en-US,en;q=0.8

HTTP Headers

```
def get(self):  
    print(self.request)  
    print(str(self.request.headers))  
    print(self.request.headers["Accept-Language"])  
  
    self.set_status(987, "Oh Yeah")  
    self.set_header("My-Personal-Value", "TOPHER")  
  
    self.write("GET requested")
```

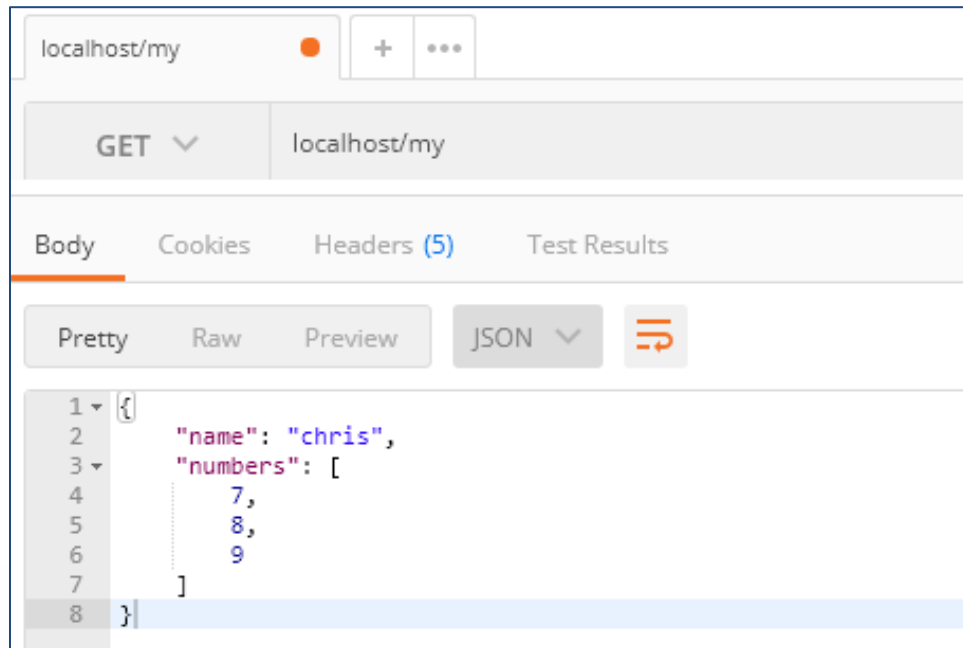
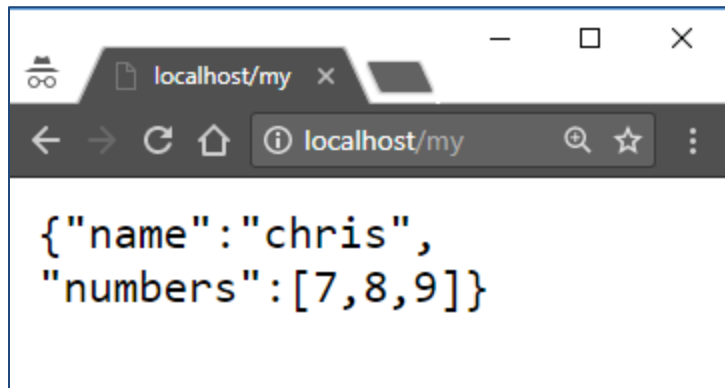


Content Type

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

```
    self.set_header("Content-Type", "application/json")
```

```
    self.write('{"name": "chris", "numbers": [7, 8, 9]}')
```

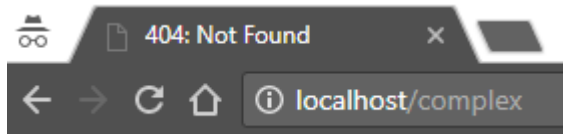


URL Decoding

```
class DecodeHandler(tornado.web.RequestHandler):  
    def get(self, first, second):  
        if (first=="chris"):  
            self.write("Hi Chris ")  
            self.write(":"+first+": "+second+":")  
  
handlers = [  
    ("/my", MyHandler),  
    ("/another", AnotherHandler),  
    ("/complex/(.*)/(.*)", DecodeHandler)  
]
```

localhost/complex/chris/789/a/b/c/d

:chris/789/a/b/c:d:



404: Not Found

localhost/complex/bob

404: Not Found

localhost/complex/bob/1234

:bob:1234:

localhost/complex/chris/789

Hi Chris :chris:789:

Query Parameters

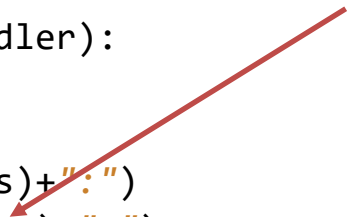


```
class AnotherHandler(tornado.web.RequestHandler):  
    def get(self):  
  
        print(": "+str(self.request.arguments)+":")  
        print(": "+self.get_argument("b")+":")  
        print(": "+str(self.get_arguments("b"))+":")  
  
        self.write("This is AnotherHandler")
```

```
{'a': [''], 'c': ['chris'], 'b': ['1', '2']}  
:2:  
:[u'1', u'2']:
```

Missing Parameters

get_argument returns this default
if the parameter is missing



```
class AnotherHandler(tornado.web.RequestHandler):  
    def get(self):  
  
        print(":"+str(self.request.arguments)+":")  
        print(":"+self.get_argument("b", None)+":")  
        print(":"+str(self.get_arguments("b"))+":")  
  
        self.write("This is AnotherHandler")
```

get_argument raises an exception
if the default is not given

```
<html>
```

```
<form action="my" method="post">
```

```
    Name: <input type="text" name="name"> <br>
```

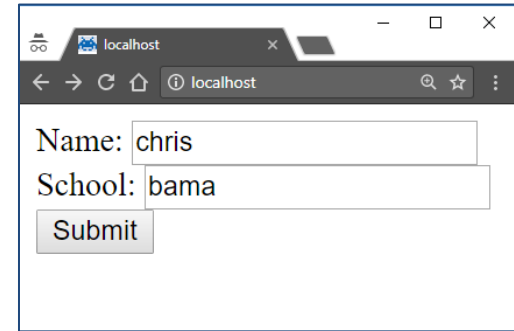
```
    School: <input type="text" name="school"> <br>
```

```
    <input type="submit">
```

```
</form>
```

```
</html>
```

Reading the Request Body



A screenshot of a web browser window showing a form. The form has two text input fields: "Name:" with the value "chris" and "School:" with the value "bama". Below the fields is a "Submit" button. The browser's address bar shows "localhost".

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

```
    def post(self):
```

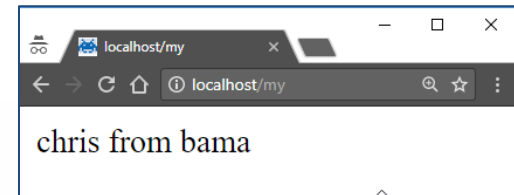
```
        print(self.request.arguments)
```

```
        print(self.request.body)
```

```
        self.write(self.get_argument("name")+" from "+  
                    self.get_argument("school"))
```

```
{'school': ['bama'], 'name': ['chris']}
```

name=chris&school=bama



A screenshot of a web browser window showing the output of the form submission. The browser's address bar shows "localhost/my". The page content displays "chris from bama".

Reading the Request Body

POST ▾

localhost/my

Params

Send ▾

Key	Value	Description
New key	Value	Description

Authorization Headers **Body** Pre-request Script Tests

☒ form-data ☐ x-www-form-urlencoded ☐ raw ☐ binary

Key	Value	Description
<input checked="" type="checkbox"/> name	chris	
<input checked="" type="checkbox"/> school	bama	
New key	Value	Description

Body Cookies Headers (4) Test Results

Status: 200 OK

Pretty Raw Preview HTML ▾ ↺

i 1 chris from bama

Tinkering

- Code up a tornado server that serves out an index.html file with a simple form in it.
- The form has two numbers and a submit button.
- The server adds the two numbers and returns the output.

