

---

# **Tortik Documentation**

***Release 0.2.3***

**The Tortik Authors**

November 18, 2016



<b>1</b>	<b>Installation</b>	<b>3</b>
<b>2</b>	<b>Documentation</b>	<b>5</b>
2.1	<code>tortik.page.RequestHandler</code> — base class for request handle . . . . .	5
	<b>Python Module Index</b>	<b>7</b>



Tortik is a frontend micro framework atop of Python Tornado making easier to develop SOA-based applications.

Short explanation of tortik features: <http://glibin.github.io/tortik>

Documentation and installation instructions <http://tortik.readthedocs.org/en/latest/>

SOA (microservices) and tortik use cases at PyCon RU 2014 (in Russian) <http://glibin.github.io/lections/pycon2014/>



---

# Installation

---

### Automatic installation:

```
pip install tortik
```

Tortik is listed in [PyPI](#) and can be installed with `pip` or `easy_install`.

If you plan to use XML backends it's highly recommended to install [lxml](#) package (tortik will use it if available) which is much faster than native implementation and have more features and additional libraries.





## 2.1 `tortik.page.RequestHandler` — base class for request handle

### 2.1.1 Base class for request handle

**class** `tortik.page.RequestHandler` (*application, request, \*\*kwargs*)

Base handler for request handle

Differs from `tornado.web.RequestHandler` that all method handlers are `@tornado.web.asynchronous` by default.

Handler completion should be done with `self.complete` method instead of `self.finish` for applying postprocessors.

### 2.1.2 RequestHandler

#### Methods

`RequestHandler.make_request` (*name, method='GET', full\_url=None, url\_prefix=None, path='', data='', headers=None, connect\_timeout=1, request\_timeout=2, follow\_redirects=True, \*\*kwargs*)

Class for easier constructing `tornado.httpclient.HTTPRequest` object.

Request url could be constructed with two ways:

- `full_url` argument
- `url_prefix` as domain part and `path` as path part

#### Parameters

- **name** (*string*) – Name of the request (for later accessing response through `self.responses.get(name)`)
- **method** (*string*) – HTTP method, e.g. “GET” or “POST”
- **full\_url** (*string*) – Full url for the requesting server (ex. `http://example.com`)
- **url\_prefix** (*string*) – Request url domain part
- **path** (*string*) – Request url path part

- **data** (*string* or *dict*) – Query to be passed to the request (could be a dict and would be translated to a query string)
- **headers** (`tornado.httputil.HTTPHeaders` or *dict*) – Additional HTTP headers to pass on the request
- **connect\_timeout** (*float*) – Timeout for initial connection in seconds
- **request\_timeout** (*float*) – Timeout for entire request in seconds
- **follow\_redirects** (*bool*) – Should redirects be followed automatically or return the 3xx response?
- **kwargs** – any other `tornado.httpclient.HTTPRequest` arguments

**Returns** `tornado.httpclient.HTTPRequest`

- [genindex](#)
- [modindex](#)
- [search](#)

**t**

`tortik.page`, 5



## M

`make_request()` (`tortik.page.RequestHandler` method), [5](#)

## R

`RequestHandler` (class in `tortik.page`), [5](#)

## T

`tortik.page` (module), [5](#)