

---

# **Vumi HTTP Proxy Documentation**

***Release 0.1.1***

**Carla Wilby**

February 12, 2016



|          |                                            |          |
|----------|--------------------------------------------|----------|
| <b>1</b> | <b>Contents:</b>                           | <b>3</b> |
| 1.1      | Project Outline . . . . .                  | 3        |
| 1.2      | Twistd ServiceMaker Plugin Usage . . . . . | 3        |
| 1.3      | Click queen-of-ni Usage . . . . .          | 4        |
| 1.4      | vumi_http_proxy Package Docs . . . . .     | 4        |
| <b>2</b> | <b>Indices and tables</b>                  | <b>7</b> |



Black & White listing HTTP Proxy for Vumi Sandbox HTTP requests



---

**Contents:**

---

## 1.1 Project Outline

Vumi Go, the hosted Vumi environment, while being horizontally scalable and consisting of a number of moving parts over time grew to be more monolithic than originally designed. We are in the process of breaking off different chunks and housing these in separate micro-services. The idea of micro-services is to make a single bit of server software responsible for a single task instead of everything being inlined in the main application. These micro-services interact with the wider system through a number of APIs and as a result should draw clearer lines of separation & responsibility between the various moving parts of a large application. The idea behind micro-services is also that they would allow for the internals to be more easily refactored and supported by specific teams.

All of our micro-services that we are running or are designing expose HTTP APIs. Generally they accept and respond with JSON payloads. These services need to be isolated from other applications. For obvious reasons we cannot allow rogue applications access internal APIs.

Vumi's Javascript sandbox allows application developers to access HTTP resources, we need a proxy that intercepts these outbound HTTP calls and checks whether the calls to the specified resources are allowed or not. This boils down to maintaining a blacklist of disallowed HTTP resources and per request checking against the blacklist.

## 1.2 Twistd ServiceMaker Plugin Usage

This plugin allows manual configuration of the HTTP Proxy.

First, build this locally:

```
$ virtualenv ve
$ source ve/bin/activate
(ve)$ pip install -e .
```

To see list of all twistd plugins:

```
(ve)$ twistd -d
```

In order to configure the proxy using default settings, a blacklist must be specified as follows:

```
(ve)$ twistd -n vumi_http_proxy --blacklist=BLACKLISTFILENAME
```

This blacklist must be a PyYAML configuration file, an example of which can be found below (`./docs/proxy_blacklist.yml`):

The default settings are:

```
IP address: 0.0.0.0
port: 8080
blacklist: None
```

To run using manual configuration:

```
(ve)$ twistd -n vumi_http_proxy --interface=IPADDRESS --port=PORTNO --blacklist=BLACKLISTFILENAME
```

For hints and a list of all available commands, please see:

```
(ve)$ twistd -n vumi_http_proxy --help
```

**Warning:** This version does not yet support HTTPS requests

## 1.3 Click queen-of-ni Usage

This plugin allows manual configuration of the HTTP Proxy (but with a cool name).

First, build this locally:

```
$ virtualenv ve
$ source ve/bin/activate
(ve)$ pip install -e .
```

In order to run queen-of-ni a blacklist file must be specified. This must be a PyYAML configuration file - an example of which can be found below (`./docs/proxy_blacklist.yml`):

The default settings are:

```
IP address: 0.0.0.0
port: 8080
blacklist: None
```

To run queen-of-ni using default configuration, run:

```
(ve)$ queen-of-ni --blacklist BLACKLISTFILENAME
```

Otherwise to use manual configuration, run:

```
(ve)$ queen-of-ni --interface IPADDRESS --port PORTNO --blacklist BLACKLISTFILENAME
```

For help, see:

```
(ve)$ queen-of-ni --help
```

**Warning:** This version does not yet support HTTPS requests

## 1.4 vumi\_http\_proxy Package Docs

Class Documentation

#### 1.4.1 http\_proxy Module

#### 1.4.2 queen-of-ni Module

#### 1.4.3 servicemaker Module

#### 1.4.4 config\_reader Module



---

## Indices and tables

---

- `genindex`
- `modindex`
- `search`