

httpbis  
Internet-Draft  
Intended status: Standards Track  
Expires: 8 January 2026

B. M. Schwartz  
Meta Platforms, Inc.  
7 July 2025

Template-Driven HTTP Request Proxying  
draft-schwartz-modern-http-proxies-02

Abstract

HTTP request proxying behaviors have long been part of the core HTTP specification. However, the core request proxying functionality has several important deficiencies in modern HTTP environments. This specification defines an alternative proxy service configuration for HTTP requests. The proxy service is identified by a URI Template, similarly to "connect-tcp" and "connect-udp".

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at <https://datatracker.ietf.org/drafts/current/>.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on 8 January 2026.

Copyright Notice

Copyright (c) 2025 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<https://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Revised BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Revised BSD License.

## Table of Contents

1. Introduction . . . . .	2
1.1. History . . . . .	2
1.2. Problems . . . . .	2
1.3. Overview . . . . .	3
2. Conventions and Definitions . . . . .	3
3. Requirements . . . . .	3
3.1. Use of Templates . . . . .	3
3.2. Configuration . . . . .	3
4. Examples . . . . .	4
5. Security Considerations . . . . .	7
6. Operational Considerations . . . . .	7
7. IANA Considerations . . . . .	7
8. References . . . . .	7
8.1. Normative References . . . . .	7
8.2. Informative References . . . . .	8
Acknowledgments . . . . .	8
Author's Address . . . . .	8

## 1. Introduction

## 1.1. History

An HTTP forward proxy (or just "proxy" in the HTTP standards) is an HTTP service that acts on behalf of the client as an intermediary for some or all HTTP requests. HTTP/1.0 defined the initial HTTP proxying mechanism: the client formats its request target in "absolute form" (i.e., with a full URI in the Request-Line) and delivers it to the proxy, which reissues the request to the origin specified in the URI ([RFC1945], Section 5.1.2). In this specification, we call this behavior a "classic HTTP request proxy".

In HTTP/1.1, proxy requests are additionally required to carry a Host header whose value matches the authority in the Request URI (not the name of the proxy server). In HTTP/2 and HTTP/3, the destination host is specified in the :authority pseudo-header field ([RFC9113], Section 8.3.1).

## 1.2. Problems

HTTP clients can be configured to use proxies by selecting a proxy host, a port, and whether to use a security protocol. However, requests to the proxy do not carry this configuration information. Instead, they only indicate the URI of the requested resource. This prevents any HTTP server from hosting multiple distinct proxy services, as the server cannot distinguish them by path (as with distinct resources) or by origin (as in "virtual hosting").

The absence of an explicit origin for the proxy also rules out the usual defenses against server port misdirection attacks (see Section 7.4 of [RFC9110]).

### 1.3. Overview

This specification describes an alternative protocol for an HTTP request proxy. Like CONNECT-TCP, CONNECT-UDP, and CONNECT-IP, the proxy is identified by a URI Template.

## 2. Conventions and Definitions

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in BCP 14 [RFC2119] [RFC8174] when, and only when, they appear in all capitals, as shown here.

## 3. Requirements

### 3.1. Use of Templates

A templated HTTP request proxy is identified by a URI Template containing a variable named "target\_uri". To convert an HTTP request into a proxied request, the client MUST substitute the request's URI into this variable, expand the template, and use the result as the new request URI.

HTTP headers and status codes are processed in the same way as in classic HTTP request proxies.

A templated HTTP request proxy is also suitable for use as an Oblivious HTTP relay, if it provides the required privacy guarantees.

### 3.2. Configuration

Clients that support both classic HTTP request proxies and template-driven proxies MAY accept both types via a single configuration string. If the configuration string can be parsed as a URI Template containing the "target\_uri" variable, it is a template-driven request proxy. Otherwise, it is presumed to represent a classic HTTP request proxy.

This specification defines a new Proxy-Status parameter:

"use\_template" (see registration in Section 7), which conveys a preferred URI Template for the proxy. Upon receipt of this parameter, the client SHOULD update its configuration to use the new template for the remainder of the session if possible, and retry the request using the new template if it also received a Proxy-Status "error" parameter.

If the client is configured with a classic HTTP request proxy, and the template string is the special value "default", the client MUST use the following default proxy template:

```
https://$PROXY_HOST:$PROXY_PORT/.well-known/masque
    /http/{target_uri}
```

Figure 1: Registered default template

This allows a virtual-hosted proxy server to learn the proxy's hostname, which is not present in the initial request.

#### 4. Examples

Consider a proxy identified as "https://example.com/proxy{?target\_uri}". Requests would then be transformed as follows:

Original request:

```
PATCH /resource HTTP/1.1
Host: api.example
Content-Type: application/example
...
```

Transformed request:

```
PATCH /proxy?target_uri=https%3A%2F%2Fapi.example%2Fresource HTTP/1.1
Host: example.com
Content-Type: application/example
Proxy-Authorization: ...
...
```

Notes on this example:

- \* The HTTP method is not altered.
- \* The request-related headers such as Content-Type are preserved, but the Host header (or :authority in HTTP/2 and HTTP/3) is altered.
- \* Certain characters in the target URI are percent-encoded during URI Template expansion.

- \* The scheme, which is implicit in the original request, is explicit in the transformed request. The scheme in this example is "https", indicating that the client is asking the proxy to establish a secure connection to the target.
- \* The client can add Proxy-\* headers to communicate with the proxy.

A templated HTTP request proxy can be used as an Oblivious HTTP Relay. For example, suppose the relay is identified as "https://proxy.example.org/relay{?target\_uri}", and the Oblivious HTTP Gateway is "https://example.com/gateway". The client would send requests to the proxy as follows:

```
POST /relay?target_uri=https%3A%2F%2Fexample.com%2Fgateway HTTP/1.1
Host: proxy.example.org
Proxy-Authorization: ...
Content-Type: message/ohhttp-req
...
```

Figure 2: Use of an HTTP request proxy as an Oblivious relay

If a templated HTTP request proxy supports HTTP/2 and Extended CONNECT, it is even possible to reach a CONNECT-TCP transport proxy through it:

```
CONNECT HTTP/2.0
:authority = request-proxy.example
:scheme = https
:path = /proxy?target_uri=https%3A%2F%2Ftransport-proxy.example%2Fproxy
      %3Ftarget_host%3Ddestination.example%26target_port%3D443
:protocol = connect-tcp
capsule-protocol: ?1
...
```

Figure 3: Use of a TCP transport proxy through an HTTP request proxy

A proxy can use the "use\_template" proxy status error to reconfigure existing clients:

```
GET /main?target_uri=https%3A%2F%2Fexample.com%2F HTTP/1.1
Host: proxy.example.org
Proxy-Authorization: ...
Content-Type: message/ohttp-req
...

HTTP/1.1 200 OK
Proxy-Status: proxy.example.org; \
  use_template="https://proxy.example.org/beta{?target_uri}"; \
  details="You have been assigned to the beta test group"
Content-Type: message/ohttp-res
...
```

Figure 4: Updating the configured template with 'use\_template'

If the client's proxy configuration string was "proxy-foo.example.org:54321", the client will start by issuing a classic HTTP proxy request, but the proxy can use the "use\_template" parameter to inform the client that it should use templated requests instead:

```
GET https://example.com/ HTTP/1.1
Proxy-Authorization: ...
Host: example.com
Accept: text/html

HTTP/1.1 400 Bad Request
Proxy-Status: proxy-no-template.example.org; \
  use_template="default"; \
  error="http_request_denied"; \
  details="Proxy template required"

GET .well-known/masque/http/https%3A%2F%2Fexample.com%2F HTTP/1.1
Proxy-Authorization: ...
Host: proxy-foo.example.org:54321
Accept: text/html

HTTP/1.1 200 OK
Content-Type: text/html
Content-Length: ...
...
```

Figure 5: Using 'use\_template' to upgrade from classic to templated proxying

## 5. Security Considerations

None

## 6. Operational Considerations

Templated HTTP proxies can make use of standard HTTP gateways and path-routing to ease implementation and allow use of shared infrastructure. To be compatible, a gateway must forward Proxy-\* request headers to the origin.

## 7. IANA Considerations

IF APPROVED, IANA is requested to add the following entry to the "MASQUE URI Suffixes" registry:

Path Segment	Description	Reference
http	HTTP Request Proxying	(This document)

Table 1

IF APPROVED, IANA is requested to add the following entry to the "HTTP Proxy Status Parameters" registry:

- \* Name: use\_template
- \* Description: A URI Template that should be used for requests to this proxy server. The special value "default" indicates that the client should use the default template for the configured proxy hostname (which is not known to the proxy).
- \* Reference: (This document)

## 8. References

### 8.1. Normative References

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <<https://www.rfc-editor.org/rfc/rfc2119>>.
- [RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <<https://www.rfc-editor.org/rfc/rfc8174>>.

## 8.2. Informative References

- [RFC1945] Berners-Lee, T., Fielding, R., and H. Frystyk, "Hypertext Transfer Protocol -- HTTP/1.0", RFC 1945, DOI 10.17487/RFC1945, May 1996, <<https://www.rfc-editor.org/rfc/rfc1945>>.
- [RFC9110] Fielding, R., Ed., Nottingham, M., Ed., and J. Reschke, Ed., "HTTP Semantics", STD 97, RFC 9110, DOI 10.17487/RFC9110, June 2022, <<https://www.rfc-editor.org/rfc/rfc9110>>.
- [RFC9113] Thomson, M., Ed. and C. Benfield, Ed., "HTTP/2", RFC 9113, DOI 10.17487/RFC9113, June 2022, <<https://www.rfc-editor.org/rfc/rfc9113>>.

## Acknowledgments

TODO acknowledge.

## Author's Address

Benjamin M. Schwartz  
Meta Platforms, Inc.  
Email: [ietf@bemasc.net](mailto:ietf@bemasc.net)