

Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: May 4, 2018

A. Hope-Bailie
Ripple
October 31, 2017

HTTP-Payments
draft-hope-bailie-http-payments-00

Abstract

HTTP-Payments describes a mechanism for passing a standardized payment request in the headers of an HTTP 402 response and the expected behaviour of HTTP clients that receive such a response.

Feedback

This specification is an early experiment in bringing the work of the W3C Web Payments working group to the HTTP protocol. It is maintained at <https://github.com/adrianhopebailie/http-payments> [1].

The work is inspired by work in the Interledger community on [HTTP-ILP]

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 May 4, 2018.

Copyright Notice

Copyright (c) 2017 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 Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1. Introduction	2
2. Terminology	3
3. Payment Methods	3
4. HTTP Status Code 402	3
4.1. The "Pay" Header	3
4.2. The "Pay-Token" Header	4
4.3. The "Pay-Balance" Header	4
4.4. Flow	4
4.5. Example	5
5. References	5
5.1. Normative References	5
5.2. Informative References	6
5.3. URIs	6
Appendix A. Security Considerations	6
Appendix B. IANA Considerations	6
B.1. Payment Method Identifier Short-string Registry	6
Author's Address	6

1. Introduction

The W3C Web Payments working group has defined a Web Platform API that is being widely deployed to browsers for requesting a payment. The PaymentRequest API [W3C.CR-payment-request-20170921], defines an interface for a website to pass a payment request to the user agent via this API.

The user agent will then, through interaction with the user, complete or reject the requested payment.

HTTP-Payments describes a manner in which an HTTP server can request payment from a client in the same manner as a website would from a user agent using the W3C APIs.

The critical portion of the payment request is the set of, one or more, supported payment methods and associated payment-method-specific data. HTTP-Payments defines a mechanism by which these are expressed in the response headers of an HTTP request for which the server requires a payment.

In the website and user-agent scenario, when handling the payment request, the user-agent will prompt the user to pick one of the supported payment methods and will then handle the payment in a manner that is appropriate for that payment method. In an HTTP-Payment, the HTTP client will perform this function, likely with no user interaction.

2. Terminology

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [RFC2119][].

3. Payment Methods

A payment method is a way that the payee can be paid. Examples include, via credit card, bank wire transfer, or Bitcoin.

A payment method is identified by a payment method identifier as specified in the Payment Method Identifiers specification [2]. This is either a standardized short-string, identified in a registry maintained by the W3C Web Payments WG, or a URL.

The most common case will be for the URL form to be used. In cases where there is no authority responsible for the payment method that can host the payment method URL, the WG will consider adding a new identifier for the payment method to the registry.

Payment methods define the data that the payer and payee need to exchange, to complete a payment, and the process by which this occurs.

4. HTTP Status Code 402

The HTTP Status Code, 402 (Payment Required) is currently defined in [RFC7231] as "reserved for future use". Using HTTP-Payment a service MAY respond to any request with the 402 response code and use the "Pay" header to specify the payment request details.

4.1. The "Pay" Header

The body of the "Pay" header is defined as follows:

Pay: <payment-method-identifier> <amount> <address> <payment-method-data>

Multiple "Pay" headers MAY be present in an HTTP 402 response.

The fields in the header are:

- o payment-method: The payment method identifier for the accepted payment method. Either a standardized short-string or a URL.
- o amount: The amount that must be paid, expressed as an integer. The currency, scale and precision of the destination account are expected to be expressed in the account address.
- o address: A payment-method specific payee address. For example, if the payment method is Bitcoin this would be a Bitcoin address.
- o payment-method-data: Payment method specific data. This is either a URI identifying the data or, if it is small enough, is the data itself, BASE64URL encoded as described in [RFC4648], Section 5.

4.2. The "Pay-Token" Header

An HTTP client that makes a paid-HTTP request, after paying for the request to be processed, MAY attach a "Pay-Token" header with a token referencing the payment.

This mechanism can be employed by services wishing to accept payments without binding these to an HTTP session.

4.3. The "Pay-Balance" Header

An HTTP Service that accepts payments may respond to any request with a "Pay-Balance" header. This contains an integer indicating the current balance of paid credit the client has with the HTTP service.

4.4. Flow

Upon receipt of a 402 response, an HTTP client MUST look for any "Pay" headers and parse these. The client can discard all headers for which it is not equipped to make a payment (i.e. filter on payment-method-identifier)

The client MUST then select the header that is preferred for processing based upon external interactions (such as with a human user) or pre-configured rules. The client MUST attempt to make a payment using the payment method identified in the header, for the amount specified, and to the destination address specified.

The payment-method specific data SHOULD be sufficient for the system processing the payment to reconcile the payment with the original HTTP request.

The client SHOULD receive a token in return for completing the payment. If the payment method used does return a token to the payer, it MUST pass this token in subsequent HTTP requests.

The token MUST be passed in the "Pay-Token" header, BASE64URL encoded as described in [RFC4648], Section 5.

The HTTP service MUST process the "Pay-Token" header and use this to reconcile this HTTP request with the payment received prior.

4.5. Example

Client requests access to a paid resource:

```
POST /upload HTTP/1.1
Host: myservice.example
```

Server responds with payment request (and optionally indicates that the client has a zero balance):

```
HTTP/1.1 402 Payment Required
Pay: http://interledger.org 10 us.nexus.ankita.~recv.filepay SkTcFTZCBKgP6A6QOUV
cwWCCgYIP4rJPHlIzreavHdU
Pay-Balance: 0
```

Client makes the payment through an appropriate payment side-channel and then attempts the request again:

```
POST /upload HTTP/1.1
Host: myservice.example
Pay-Token: 7y0SfeN7lCuq0GFF5UsMYZofIjJ7LrvPvsePVWSv450
```

Server responds:

```
HTTP/1.1 200 Success
Pay-Balance: 0
```

5. References

5.1. Normative References

- [RFC4648] Josefsson, S., "The Base16, Base32, and Base64 Data Encodings", RFC 4648, DOI 10.17487/RFC4648, October 2006, <<https://www.rfc-editor.org/info/rfc4648>>.
- [RFC7231] Fielding, R., Ed. and J. Reschke, Ed., "Hypertext Transfer Protocol (HTTP/1.1): Semantics and Content", RFC 7231, DOI 10.17487/RFC7231, June 2014, <<https://www.rfc-editor.org/info/rfc7231>>.

[W3C.CR-payment-method-id-20170914]

Bateman, A., Koch, Z., McElmurry, R., and M. Caceres,
"Payment Method Identifiers", World Wide Web Consortium
CR CR-payment-method-id-20170914, September 2017,
<[https://www.w3.org/TR/2017/
CR-payment-method-id-20170914](https://www.w3.org/TR/2017/CR-payment-method-id-20170914)>.

[W3C.CR-payment-request-20170921]

Bateman, A., Koch, Z., McElmurry, R., Denicola, D., and M.
Caceres, "Payment Request API", World Wide Web Consortium
CR CR-payment-request-20170921, September 2017,
<<https://www.w3.org/TR/2017/CR-payment-request-20170921>>.

5.2. Informative References

[HTTP-ILP]

Interledger Community Group, "HTTP-ILP", October 2017,
<[https://github.com/interledger/rfcs/
blob/58d8dcb015b160a381313126fa3065c64406db05/0014-http-
ilp/0014-http-ilp.md](https://github.com/interledger/rfcs/blob/58d8dcb015b160a381313126fa3065c64406db05/0014-http-ilp/0014-http-ilp.md)>.

5.3. URIs

[1] <https://github.com/adrianhopebailie/http-payments>

[2] W3C.CR-payment-method-id-20170914

Appendix A. Security Considerations

TBD

Appendix B. IANA Considerations

B.1. Payment Method Identifier Short-string Registry

The W3C maintains a registry of standardized short-string payment method identifiers as part of the [Payment Method Identifier] specification. If standardized short-string identifiers are to be used for HTTP-Payments this may be better served as an IANA registry.

Author's Address

Adrian Hope-Bailie
Ripple
315 Montgomery Street
San Francisco, CA 94104
US

Phone: -----
Email: adrian@ripple.com
URI: <https://www.ripple.com>