

Network Working Group  
Internet-Draft  
Intended status: Standards Track  
Expires: November 15, 2020

D. Fuelling  
Ripple  
May 14, 2020

The 'payid' URI Scheme  
draft-fuelling-payid-uri-01

## Abstract

This specification defines the 'payid' Uniform Resource Identifier (URI) scheme as a way to identify a payment account at a service provider, irrespective of the particular protocols that can be used to interact with the account.

## Feedback

This specification is a part of the PayID Protocol [1] work. Feedback related to this specification should be sent to [payid@ripple.com](mailto:payid@ripple.com) [2].

## 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 November 15, 2020.

## Copyright Notice

Copyright (c) 2020 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

Fuelling	Expires November 15, 2020	[Page 1]
Internet-Draft	The 'payid' URI Scheme	May 2020

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. Definition . . . . . 3

4. Examples . . . . . 5

5. Security Concerns . . . . . 5

6. Internationalization Concerns . . . . . 5

7. IANA Considerations . . . . . 6

8. Acknowledgements . . . . . 7

9. References . . . . . 7

    9.1. Normative References . . . . . 7

    9.2. Informative References . . . . . 8

    9.3. URIs . . . . . 8

Author's Address . . . . . 9

1. Introduction

Various Uniform Resource Identifier (URI) schemes enable interaction with, or identify resources associated with, a user account at a service provider. However, no standard identifier exists to identify a user's payment account at a service provider.

While popular URIs could be re-used as payment account identifiers, these identifiers are insufficient because they are typically recognized as supporting functionality unique to those schemes. For example, the 'mailto' scheme [RFC6068] (which enables interaction with a user's email account) is broadly deployed for messaging. Re-using this identifier for payments would likely cause confusion because one desirable quality of a payment account identifier is that it expressly does not support messaging, in order to avoid spam and/or other security concerns such as phishing attacks.

Deploying payment protocols on top of identifiers that are commonly employed for other use-cases would likely be a mis-use of those identifiers, and could also cause confusion for end-users, among other problems.

Instead, the "payid" scheme uses a new type of identifier that is intended to identify accounts for payment use-cases only.

2. Terminology

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 [RFC2119].

3. Definition

The syntax of the 'payid' URI scheme is defined under Section 7 of this document. Although 'payid' URIs take the form "user\$host", the scheme is designed for the purpose of identification instead of interaction (regarding this distinction, see Section 1.2.2 of

[RFC3986]). The "Internet resource" identified by a 'payid' URI is a user's payment account hosted at a service provider, where the service provider is typically associated with a DNS domain name. Thus, a particular 'payid' URI is formed by setting the "user" portion to the user's payment account name at the service provider and by setting the "host" portion to the DNS domain name of the service provider.

Consider the case of a user with an account name of "apollo" on a wallet service "wallet.example.com". It is taken as convention that the string "apollo\$wallet.example.com" designates that payment account. This is expressed as a URI using the 'payid' scheme as 'payid:apollo\$wallet.example.com'.

A possible scenario is for a user to register with a payment service provider using an identifier (such as an email address) that is associated with some other service provider. For example, a user with the email address "alice@example.net" might register with a wallet website whose domain name is "wallet.example.com". In order to facilitate payments to/from alice, the wallet service provider might offer alice a PayID using alice's email address (though using an email address as a PayID is not recommended). In order to use her email address as the 'acctpart' of the 'payid' URI, no percent-encoding is necessary because the 'acctpart' portion of a PayID allows for at-signs. Thus, the resulting 'payid' URI would be "payid:alice@example.net\$shoppingsite.example".

Another possible scenario is a payment service provider (e.g., a digital wallet) that allows a user to use a PayID that is associated with some other payment service provider. For example, a user with the PayID "alice\$bank.example.net" might register with a wallet website whose domain name is "wallet.example.net". In order to use her bank's PayID as the localpart of the wallet's 'payid' URI, no percent-encoding is necessary because the 'acctpart' portion of a

Fuelling	Expires November 15, 2020	[Page 3]
Internet-Draft	The 'payid' URI Scheme	May 2020

PayID allows for dollar-signs. Thus, the resulting 'payid' URI would be "payid:alice\$bank.example\$wallet.example".

It is not assumed that an entity will necessarily be able to interact with a user's PayID using any particular application protocol, such as a wallet or banking application. To enable interactions (payments or otherwise), an entity would need to use the appropriate URI scheme for such a protocol. While it might be true that the 'payid' URI minus the scheme name (e.g., "user\$example.com" derived from "payid:user\$example.com") could be used via some application protocol, that fact would be purely contingent and dependent upon the deployment practices of the payment account provider.

Because a 'payid' URI enables abstract identification only and not interaction, this specification provides no method for dereferencing a 'payid' URI on its own, e.g., as the value of the 'href' attribute of an HTML anchor element. For example, there is no behavior specified in this document for a 'payid' URI used as follows:

```
"<a href='payid:bob$example.com'>find out more</a> "
```

Any protocol that uses 'payid' URIs is responsible for specifying how

a 'payid' URI is employed in the context of that protocol (in particular, how it is dereferenced or resolved; see [RFC3986]). As a concrete example, an "Account Information" application of the WebFinger protocol [RFC7033] might take a 'payid' URI, resolve the host portion to find a WebFinger server, and then pass the 'payid' URI as a parameter in a WebFinger HTTP request for metadata (i.e., web links [RFC5988]) about the resource. For example:

```
"GET /.well-known/webfinger?resource=payid%3Abob%24example.com
HTTP/1.1 "
```

In the above example, the service retrieves the metadata associated with the payment account identified by that URI and then provides that metadata to the requesting entity in an HTTP response.

If an application needs to compare two 'payid' URIs (e.g., for purposes of authentication and authorization), it MUST do so using case normalization and percent-encoding normalization as specified in Sections 6.2.2.1 and 6.2.2.2 of [RFC3986]. In addition, the 'acctpart' is case-insensitive and therefore should be normalized to lowercase. For example, the URI "PAYID:ALICE\$www.EXAMPLE.com" is equivalent to "payid:alice\$www.example.com".

Fuelling Expires November 15, 2020 [Page 4]

Internet-Draft The 'payid' URI Scheme May 2020

#### 4. Examples

The following example URIs illustrate several variations of PayIDs and their common syntax components:

```
payid:alice$example.net
```

```
payid:john.doe$example.net
```

```
payid:jane-doe$example.net
```

#### 5. Security Concerns

Because the 'payid' URI scheme does not directly enable interaction with a user's payment account at a service provider, direct security concerns are minimized.

However, a 'payid' URI does provide proof of existence of the payment account; this implies that harvesting published 'payid' URIs could prove useful for certain attackers - for example, if an attacker can use a 'payid' URI to leverage more information about the account (e.g., via WebFinger) or if they can interact with protocol-specific URIs (such as 'mailto' URIs) whose user@host portion is the same as that of the 'payid' URI (e.g., replacing the "\$" character with an "@" sign).

In addition, protocols that make use of 'payid' URIs are responsible for defining security considerations related to such usage, e.g., the risks involved in dereferencing a 'payid' URI, the authentication and authorization methods that could be used to control access to personal data associated with a user's payment account at a service,

and methods for ensuring the confidentiality of such information.

The use of percent-encoding allows a wider range of characters in payment account names but introduces some additional risks. Implementers are advised to disallow percent-encoded characters or sequences that would (1) result in space, null, control, or other characters that are otherwise forbidden, (2) allow unauthorized access to private data, or (3) lead to other security vulnerabilities.

## 6. Internationalization Concerns

As specified in [RFC3986], the 'payid' URI scheme allows any character from the Unicode repertoire [UNICODE] encoded as UTF-8 [RFC3629] and then percent-encoded into valid ASCII [RFC0020]. Before applying any percent-encoding, an application MUST ensure the

Fuelling	Expires November 15, 2020	[Page 5]
Internet-Draft	The 'payid' URI Scheme	May 2020

following about the string that is used as input to the URI-construction process:

- o The 'acctpart' consists only of Unicode code points that conform to the PRECIS IdentifierClass specified in [RFC7564].
- o The 'host' consists only of Unicode code points that conform to the rules specified in [RFC5892].
- o Internationalized domain name (IDN) labels are encoded as A-labels [RFC5890].

## 7. IANA Considerations

In accordance with the guidelines and registration procedures for new URI schemes [RFC7595], this section provides the information needed to register the 'payid' URI scheme.

\*URI Scheme Name\*: payid

\*Status\*: permanent

\*URI Scheme Syntax\*: The 'payid' URI syntax is defined here in Augmented Backus-Naur Form (ABNF) [RFC5234], borrowing the 'host', 'pct-encoded', 'sub-delims', and 'unreserved' rules from [RFC3986]:

```
payidURI  = "payid" ":" acctpart "$" host
acctpart  = unreserved / sub-delims
           0*( unreserved / pct-encoded / sub-delims )
```

Note that additional rules regarding the strings that are used as input to construction of 'payid' URIs further limit the characters that can be percent-encoded; see the Encoding Considerations as well as Section 6 of this document.

\*URI Scheme Semantics\*: The 'payid' URI scheme identifies payment accounts hosted at payment service providers. It is used only for identification, not interaction. A protocol that employs the 'payid' URI scheme is responsible for specifying how a 'payid' URI is dereferenced in the context of that protocol. There is no media type

associated with the 'payid' URI scheme.

**\*Encoding Considerations\*:** See Section 6 of this document.

**\*Applications/Protocols That Use This URI Scheme Name\*:** At the time of this writing, only the [PAYID-DISCOVERY] protocol uses the 'payid' URI scheme. However, use is not restricted to this protocol, and the scheme might be considered for use in other protocols.

Fuelling	Expires November 15, 2020	[Page 6]
Internet-Draft	The 'payid' URI Scheme	May 2020

**\*Interoperability Considerations\*:** There are no known interoperability concerns related to use of the 'payid' URI scheme.

**\*Security Considerations\*:** See Section 5 of this document.

**\*Contact\*:** fuelling@ripple.com

**\*Author/Change Controller\*:** This scheme is registered under the IETF tree. As such, the IETF maintains change control.

**\*References\*:** None.

## 8. Acknowledgements

This document was heavily influenced RFC-7565, adapted for a payments use -case. The author would like to acknowledge the contributions of everyone who worked on that and related specifications.

In addition, the author would like to acknowledge everyone who provided feedback and use-cases for this derivative specification.

## 9. References

### 9.1. Normative References

- [PAYID-DISCOVERY]  
Fuelling, D., "The PayID Discovery Protocol", n.d.,  
<<https://tbd.example.com/>>.
- [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/info/rfc2119>>.
- [RFC3629] Yergeau, F., "UTF-8, a transformation format of ISO 10646", STD 63, RFC 3629, DOI 10.17487/RFC3629, November 2003, <<https://www.rfc-editor.org/info/rfc3629>>.
- [RFC3986] Berners-Lee, T., Fielding, R., and L. Masinter, "Uniform Resource Identifier (URI): Generic Syntax", STD 66, RFC 3986, DOI 10.17487/RFC3986, January 2005, <<https://www.rfc-editor.org/info/rfc3986>>.
- [RFC5234] Crocker, D., Ed. and P. Overell, "Augmented BNF for Syntax Specifications: ABNF", STD 68, RFC 5234, DOI 10.17487/RFC5234, January 2008, <<https://www.rfc-editor.org/info/rfc5234>>.

Fuelling Expires November 15, 2020 [Page 7]  
Internet-Draft The 'payid' URI Scheme May 2020

- [RFC5890] Klensin, J., "Internationalized Domain Names for Applications (IDNA): Definitions and Document Framework", RFC 5890, DOI 10.17487/RFC5890, August 2010, <<https://www.rfc-editor.org/info/rfc5890>>.
- [RFC5892] Faltstrom, P., Ed., "The Unicode Code Points and Internationalized Domain Names for Applications (IDNA)", RFC 5892, DOI 10.17487/RFC5892, August 2010, <<https://www.rfc-editor.org/info/rfc5892>>.
- [RFC7564] Saint-Andre, P. and M. Blanchet, "PRECIS Framework: Preparation, Enforcement, and Comparison of Internationalized Strings in Application Protocols", RFC 7564, DOI 10.17487/RFC7564, May 2015, <<https://www.rfc-editor.org/info/rfc7564>>.
- [UNICODE] The Unicode Consortium, "The Unicode Standard", n.d., <<http://www.unicode.org/versions/latest/>>.

## 9.2. Informative References

- [RFC0020] Cerf, V., "ASCII format for network interchange", STD 80, RFC 20, DOI 10.17487/RFC0020, October 1969, <<https://www.rfc-editor.org/info/rfc20>>.
- [RFC5988] Nottingham, M., "Web Linking", RFC 5988, DOI 10.17487/RFC5988, October 2010, <<https://www.rfc-editor.org/info/rfc5988>>.
- [RFC6068] Duerst, M., Masinter, L., and J. Zawinski, "The 'mailto' URI Scheme", RFC 6068, DOI 10.17487/RFC6068, October 2010, <<https://www.rfc-editor.org/info/rfc6068>>.
- [RFC7033] Jones, P., Salgueiro, G., Jones, M., and J. Smarr, "WebFinger", RFC 7033, DOI 10.17487/RFC7033, September 2013, <<https://www.rfc-editor.org/info/rfc7033>>.
- [RFC7595] Thaler, D., Ed., Hansen, T., and T. Hardie, "Guidelines and Registration Procedures for URI Schemes", BCP 35, RFC 7595, DOI 10.17487/RFC7595, June 2015, <<https://www.rfc-editor.org/info/rfc7595>>.

## 9.3. URIs

- [1] <https://payid.org/>
- [2] <mailto:payid@ripple.com>

Fuelling Expires November 15, 2020 [Page 8]  
Internet-Draft The 'payid' URI Scheme May 2020

Author's Address

David Fuelling  
Ripple  
315 Montgomery Street  
San Francisco, CA 94104  
US

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