

Independent Stream
Internet-Draft
Intended status: Informational
Expires: October 30, 2020

F. Dold
Taler Systems SA
C. Grothoff
BFH
April 28, 2020

The 'payto' URI scheme for payments draft-dold-payto-13

Abstract

This document defines the 'payto' Uniform Resource Identifier (URI) scheme for designating targets for payments.

A unified URI scheme for all payment target types allows applications to offer user interactions with URIs that represent payment targets, simplifying the introduction of new payment systems and applications.

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 October 30, 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 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
1.1.	Objective	2
1.2.	Requirements Language	3
2.	Syntax of a 'payto' URI	3
3.	Semantics	3
4.	Examples	4
5.	Generic Options	4
6.	Internationalization and Character Encoding	5
7.	Tracking Payment Target Types	5
7.1.	ACH Bank Account	6
7.2.	Business Identifier Code	7
7.3.	International Bank Account Number	7
7.4.	Unified Payments Interface	7
7.5.	Bitcoin Address	8
7.6.	Interledger Protocol Address	8
7.7.	Void Payment Target	8
8.	Security Considerations	8
9.	IANA Considerations	9
9.1.	URI Scheme Registration	9
10.	Payment Target Types	9
11.	References	10
11.1.	Normative References	10
11.2.	Informational References	11
	Authors' Addresses	11

[1.](#) Introduction

This document defines the 'payto' Uniform Resource Identifier (URI) [[RFC3986](#)] scheme for designating transfer form data for payments.

[1.1.](#) Objective

A 'payto' URI always identifies the target of a payment. A 'payto' URI consists of a payment target type, a target identifier and optional parameters such as an amount or a payment reference.

The interpretation of the target identifier is defined by the payment target type, and typically represents either a bank account or an (unsettled) transaction.

A unified URI scheme for all payment target types allows applications to offer user interactions with URIs that represent payment targets, simplifying the introduction of new payment systems and applications.

1.2. Requirements Language

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.

2. Syntax of a 'payto' URI

This document uses the Augmented Backus-Naur Form (ABNF) of [\[RFC5234\]](#).

```
payto-URI = "payto://" authority path-abempty [ "?" opts ]
opts = opt *( "&" opt )
opt-name = generic-opt / authority-specific-opt
opt-value = *pchar
opt = opt-name "=" opt-value
generic-opt = "amount" / "receiver-name" / "sender-name" /
              "message" / "instruction"
authority-specific-opt = ALPHA *( ALPHA / DIGIT / "-" / "." )
authority = ALPHA *( ALPHA / DIGIT / "-" / "." )
```

'path-abempty' is defined in [\[RFC3986\]](#) in [Section 3.3](#). 'pchar' is defined in [\[RFC3986\]](#), [Appendix A](#).

3. Semantics

The authority component of a payment URI identifies the payment target type. The payment target types are defined in the "Payment Target Types" sub-registry, see [Section 10](#). The path component of the URI identifies the target for a payment as interpreted by the respective payment target type. The query component of the URI can provide additional parameters for a payment. Every payment target type SHOULD accept the options defined in generic-opt. The default operation of applications that invoke a URI with the payto scheme MUST be to launch an application (if available) associated with the payment target type that can initiate a payment. If multiple handlers are registered for the same payment target type, the user SHOULD be able to choose which application to launch. This allows users with multiple bank accounts (each accessed the respective bank's banking application) to choose which account to pay with. An application SHOULD allow dereferencing a payto URI even if the payment target type of that URI is not registered in the "Payment Target Types" sub-registry. Details of the payment MUST be taken from the path and options given in the URI. The user SHOULD be allowed to modify these details before confirming a payment.

4. Examples

payto://iban/DE75512108001245126199?amount=EUR:200.0&message=hello

INVALID (authority missing): payto:iban/12345

5. Generic Options

Applications MUST accept URIs with options in any order. The "amount" option MUST NOT occur more than once. Other options MAY be allowed multiple times, with further restrictions depending on the payment target type. The following options SHOULD be understood by every payment target type.

amount: The amount to transfer. The format MUST be:

```
amount = currency ":" unit [ "." fraction ]
currency = 1*ALPHA
unit = 1*(DIGIT / ",")
fraction = 1*(DIGIT / ",")
```

If a 3-letter 'currency' is used, it MUST be an [\[ISO4217\]](#) alphabetic code. A payment target type MAY define semantics beyond ISO 4217 for currency codes that are not 3 characters. The 'unit' value MUST be smaller than 2⁵³. If present, the 'fraction' MUST consist of no more than 8 decimal digits. The use of commas is optional for readability and they MUST be ignored.

receiver-name: Name of the entity that receives the payment (creditor). The value of this option MAY be subject to lossy conversion, modification and truncation (for example, due to line wrapping or character set conversion).

sender-name: Name of the entity that makes the payment (debtor). The value of this option MAY be subject to lossy conversion, modification and truncation (for example, due to line wrapping or character set conversion).

message: A short message to identify the purpose of the payment. The value of this option MAY be subject to lossy conversion, modification and truncation (for example, due to line wrapping or character set conversion).

instruction: A short message giving payment reconciliation instructions to the recipient. An instruction that follows the

character set and length limitation defined by the respective payment target type SHOULD NOT be subject to lossy conversion.

6. Internationalization and Character Encoding

Various payment systems use restricted character sets. An application that processes 'payto' URIs MUST convert characters that are not allowed by the respective payment systems into allowable character using either an encoding or a replacement table. This conversion process MAY be lossy, except for the instruction field. If the value of the instruction field would be subject to lossy conversion, modification or truncation, the application SHOULD refuse further processing of the payment until a different value for the instruction is provided.

To avoid special encoding rules for the payment target identifier, the userinfo component [[RFC3986](#)] is disallowed in payto URIs. Instead, the payment target identifier is given as an option, where encoding rules are uniform for all options.

Defining a generic way of tagging the language of option fields containing natural language text (such as "receiver-name", "sender-name" and "message") is out of the scope of this document, as internationalization must accomodate the restrictions and requirements of the underlying banking system of the payment target type. The internationalization concerns SHOULD be individually defined by each payment target type.

7. Tracking Payment Target Types

A registry of Payment Target Types is described in [Section 10](#). The registration policy for this registry is "First Come First Served", as described in [[RFC8126](#)]. When requesting new entries, careful consideration of the following criteria is strongly advised:

1. The description clearly defines the syntax and semantics of the payment target and optional parameters if applicable.
2. Relevant references are provided if they are available.
3. The chosen name is appropriate for the payment target type, does not conflict with well-known payment systems, and avoids potential to confuse users.
4. The payment system underlying the payment target type is not fundamentally incompatible with the general options (such as positive decimal amounts) in this specification.

5. The payment target type is not a vendor-specific version of a payment target type that could be described more generally by a vendor-neutral payment target type.
6. The specification of the new payment target type remains within the scope of payment transfer form data. In particular specifying complete invoices is not in scope. Neither are processing instructions to the payment processor or bank beyond a simple payment.
7. The payment target and the options do not contain the payment sender's account details.

Documents that support requests for new registry entries should provide the following information for each entry:

- o Name: The name of the payment target type (case insensitive ASCII string, restricted to alphanumeric characters, dots and dashes)
- o Description: A description of the payment target type, including the semantics of the path in the URI if applicable.
- o Example: At least one example URI to illustrate the payment target type.
- o Contact: The contact information of a person to contact for further information
- o References: Optionally, references describing the payment target type (such as an RFC) and target-specific options, or references describing the payment system underlying the payment target type.

This document populates the registry with six entries as follows (see also [Section 10](#)).

7.1. ACH Bank Account

- o Name: ach
- o Description: Automated Clearing House. The path consist of two components, the routing number and the account number.
- o Example: payto://ach/122000661/1234
- o Contact: N/A
- o References: [[NACHA](#)], [this.I-D]

7.2. Business Identifier Code

- o Name: bic
- o Description: Business Identifier Code. The path consist of just a BIC. This is used for wire transfers between banks. The registry for BICs is provided by SWIFT. The path does not allow specifying a bank account number.
- o Example: `payto://bic/SOGEDEFFXXX`
- o Contact: N/A
- o References: [[BIC](#)], [this.I-D]

7.3. International Bank Account Number

- o Name: iban
- o Description: International Bank Account Number (IBAN). Generally the IBAN allows to unambiguously derive the the associated Business Identifier Code (BIC). However, some legacy applications process payments to the same IBAN differently based on the specified BIC. Thus the path can either consist of a single component (the IBAN) or two components (BIC followed by IBAN).
- o Example: `payto://iban/DE75512108001245126199`
`payto://iban/SOGEDEFFXXX/DE75512108001245126199`
- o Contact: N/A
- o References: [[ISO20022](#)], [this.I-D]

7.4. Unified Payments Interface

- o Name: upi
- o Description: Unified Payment Interface. The path is an account alias. The amount and receiver-name options are mandatory for this payment target.
- o Example: `payto://upi/alice@example.com?receiver-name=Alice&amount=INR:200`
- o Contact: N/A
- o References: [[UPILinking](#)], [this.I-D]

7.5. Bitcoin Address

- o Name: bitcoin
- o Description: Bitcoin protocol. The path is a "bitcoinaddress" as per [\[BIP0021\]](#).
- o Example: `payto://bitcoin/12A1MyfXbW6RhdRAZEqofac5jCQQjwEPBu`
- o Contact: N/A
- o References: [\[BIP0021\]](#), [this.I-D]

7.6. Interledger Protocol Address

- o Name: ilp
- o Description: Interledger protocol. The path is an ILP address as per [\[ILP-ADDR\]](#).
- o Example: `payto://ilp/g.acme.bob`
- o Contact: N/A
- o References: [\[ILP-ADDR\]](#), [this.I-D]

7.7. Void Payment Target

- o Name: void
- o Description: The "void" payment target type allows specifying the parameters of an out-of-band payment (such as cash or other types of in-person transactions). The path is optional and interpreted as a comment.
- o Example: `payto://void/?amount=EUR:10.5`
- o Contact: N/A
- o References: [this.I-D]

8. Security Considerations

Interactive applications handling the payto URI scheme MUST NOT initiate any financial transactions without prior review and confirmation from the user, and MUST take measures to prevent clickjacking [\[HMW12\]](#).

Unless a payto URI is received over a trusted, authenticated channel, a user might not be able to identify the target of a payment. In particular due to homographs [[unicode-tr36](#)], a payment target type SHOULD NOT use human-readable names in combination with unicode in the target account specification, as it could give the user the illusion of being able to identify the target account from the URI.

The authentication/authorization mechanisms and transport security services used to process a payment encoded in a payto URI are handled by the application and are not in scope of this document.

To avoid unnecessary data collection, payment target types SHOULD NOT include personally identifying information about the sender of a payment that is not essential for an application to conduct a payment.

9. IANA Considerations

IANA maintains a registry called the "Uniform Resource Identifier (URI) Schemes" registry.

9.1. URI Scheme Registration

IANA maintains the "Uniform Resource Identifier (URI) Schemes" registry that contains an entry for the 'payto' URI scheme. IANA is requested to update that entry to reference this document when published as an RFC.

10. Payment Target Types

This document specifies a list of Payment Target Types. It is possible that future work will need to specify additional payment target types. The GNUnet Assigned Numbers Authority (GANA) [[GANA](#)] operates the "payto-payment-target-types" registry to track the following information for each payment target type:

- o Name: The name of the payment target type (case insensitive ASCII string, restricted to alphanumeric characters, dots and dashes)
- o Contact: The contact information of a person to contact for further information
- o References: Optionally, references describing the payment target type (such as an RFC) and target-specific options, or references describing the payment system underlying the payment target type.

The entries that have been made for the "payto-payment-target-types" defined in this document are as follows:

Name	Contact	Reference
ach	N/A	[This.I-D]
bic	N/A	[This.I-D]
iban	N/A	[This.I-D]
upi	N/A	[This.I-D]
bitcoin	N/A	[This.I-D]
ilp	N/A	[This.I-D]
void	N/A	[This.I-D]

11. References

11.1. Normative References

- [ISO20022] International Organization for Standardization, "ISO 20022 Financial Services - Universal financial industry message scheme", May 2013.
- [ISO4217] International Organization for Standardization, "ISO 4217 Currency Codes", August 2018.
- [NACHA] NACHA, "NACHA Operating Rules & Guidelines", January 2017.
- [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>>.
- [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>>.
- [RFC8126] Cotton, M., Leiba, B., and T. Narten, "Guidelines for Writing an IANA Considerations Section in RFCs", [BCP 26](#), [RFC 8126](#), DOI 10.17487/RFC8126, June 2017, <<https://www.rfc-editor.org/info/rfc8126>>.
- [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/info/rfc8174>>.

[unicode-tr36]

Davis, M., Ed. and M. Suignard, "Unicode Technical Report #36: Unicode Security Considerations", September 2014.

11.2. Informational References

- [BIC] International Organization for Standardization, "ISO 9362:2014 Business Identifier Code (BIC)", March 2019, <<https://www.iso.org/standard/60390.html>>.
- [BIP0021] Schneider, N. and M. Corallo, "Bitcoin Improvement Proposal 21", January 2012, <https://en.bitcoin.it/wiki/BIP_0021>.
- [GANA] GNUnet e.V., "GNUnet Assigned Numbers Authority (GANA)", April 2020, <<https://gana.gnunet.org/>>.
- [HWW12] Huang, L., Moshchuk, A., Wang, H., Schecter, S., and C. Jackson, "Clickjacking: Attacks and Defenses", January 2012, <<https://www.usenix.org/system/files/conference/usenixsecurity12/sec12-final39.pdf>>.
- [ILP-ADDR] Interledger Team, "ILP Addresses - v2.0.0", September 2018, <<https://interledger.org/rfcs/0015-ilp-addresses/>>.
- [UPILinking] National Payment Corporation of India, "Unified Payment Interface - Common URL Specifications For Deep Linking And Proximity Integration", November 2017, <https://www.npci.org.in/sites/default/files/UPI%20Linking%20Specs_ver%201.6.pdf>.

Authors' Addresses

Florian Dold
Taler Systems SA
7, rue de Mondorf
Erpeldange L-5421
LU

Email: dold@taler.net

Christian Grothoff
BFH
Hoeheweg 80
Biel/Bienne CH-2501
CH

Email: christian.grothoff@bfh.ch