

Package: shortuuid (via r-universe)

August 23, 2024

Title Generate and Translate Standard UUIDs

Version 0.0.0.9000

Description Generate and translate standard UUIDs into shorter - or just different - formats and back. Also implements base58 encoders and decoders.

License MIT + file LICENSE

Encoding UTF-8

Roxygen list(markdown = TRUE)

RoxygenNote 7.3.1

LinkingTo Rcpp

Imports Rcpp

Suggests testthat (>= 3.0.0)

Config/testthat/edition 3

Repository <https://schochastics.r-universe.dev>

RemoteUrl <https://github.com/schochastics/shortuuid>

RemoteRef HEAD

RemoteSha b0a0965a5feab31570deeb4dcb74ee503eb2f077

Contents

base58_to_uuid	2
bitcoin58_to_uuid	2
flickr58_to_uuid	3
generate_uuid	3
is.base58	4
is.uuid	4
uuid_to_base58	4
uuid_to_bitcoin58	5
uuid_to_flickr58	5
validate.uuid	6

Index	7
--------------	----------

base58_to_uuid *Convert base58 to uuid*

Description

Convert base58 to uuid

Usage

```
base58_to_uuid(input, alphabet)
```

Arguments

input character vector of base58 strings
alphabet character vector representing an alphabet

Value

character vector of uuids

bitcoin58_to_uuid *Convert base58 bitcoin encoded character vector to uuid*

Description

Convert base58 bitcoin encoded character vector to uuid

Usage

```
bitcoin58_to_uuid(input)
```

Arguments

input character vector of base58 strings

Value

character vector of uuids

flickr58_to_uuid	<i>Convert base58 flickr encoded character vector to uuid</i>
------------------	---

Description

Convert base58 flickr encoded character vector to uuid

Usage

```
flickr58_to_uuid(input)
```

Arguments

input	character vector of base58 strings
-------	------------------------------------

Value

character vector of uuids

generate_uuid	<i>Generate a random RFC4122 v4-compliant UUID</i>
---------------	--

Description

Generate a random RFC4122 v4-compliant UUID

Usage

```
generate_uuid(n = 1)
```

Arguments

n	number of ids to generate
---	---------------------------

Value

character vector of uuids

<code>is.base58</code>	<i>validate if character vector is base58 encoded</i>
------------------------	---

Description

validate if character vector is base58 encoded

Usage

```
is.base58(x, alphabet)
```

Arguments

<code>x</code>	A character vector
<code>alphabet</code>	character vector representing an alphabet

<code>is.uuid</code>	<i>check if object is of class uuid</i>
----------------------	---

Description

check if object is of class uuid

Usage

```
is.uuid(x)
```

Arguments

<code>x</code>	A character vector
----------------	--------------------

<code>uuid_to_base58</code>	<i>Convert uuid to base58</i>
-----------------------------	-------------------------------

Description

Convert uuid to base58

Usage

```
uuid_to_base58(input, alphabet)
```

Arguments

input character vector of uuids
alphabet character vector representing an alphabet

Value

character vector of base58 encoded uuids

uuid_to_bitcoin58 *Convert uuid to base58 encoding of bitcoin*

Description

Convert uuid to base58 encoding of bitcoin

Usage

uuid_to_bitcoin58(input)

Arguments

input character vector of uuids

Value

character vector of base58 encoded uuids

uuid_to_flickr58 *Convert uuid to base58 encoding of flickr*

Description

Convert uuid to base58 encoding of flickr

Usage

uuid_to_flickr58(input)

Arguments

input character vector of uuids

Value

character vector of base58 encoded uuids

validate.uuid	<i>validate if a string is a uuid</i>
---------------	---------------------------------------

Description

validate if a string is a uuid

Usage

```
validate.uuid(x)
```

Arguments

x	A character vector
---	--------------------

Index

base58_to_uuid, [2](#)
bitcoin58_to_uuid, [2](#)

flickr58_to_uuid, [3](#)

generate_uuid, [3](#)

is.base58, [4](#)
is.uuid, [4](#)

uuid_to_base58, [4](#)
uuid_to_bitcoin58, [5](#)
uuid_to_flickr58, [5](#)

validate.uuid, [6](#)