# Package 'trustedtimestamping'

October 14, 2022

**Title** Create Trusted Timestamps of Datasets and Files

**Description** Trusted Timestamps (tts) are created by incorporating a hash of a file or dataset into a transaction on the decentralized blockchain (Stellar network). The package makes use of a free service provided by <https://stellarapi.io>.

**Version** 0.2.6

**License** AGPL-3

**Encoding** UTF-8

**LazyData** true

**RoxygenNote** 6.1.1

**Depends** R (>= 3.0.0)

**Imports** digest, jsonlite, httr

**BugReports** https://github.com/ttspackage/tts/issues

**NeedsCompilation** no

**Author** Peter Muller [aut, cre] (<https://orcid.org/0000-0002-5748-6270>)

**Maintainer** Peter Muller <ttspackage@gmail.com>

**Repository** CRAN

**Date/Publication** 2019-07-30 21:50:02 UTC

## R topics documented:

`convert_stellarHash` *Convert hash on STELLAR network (base64 encoded) to standard hexadecimal value*

## Description

Convert hash on STELLAR network (base64 encoded) to standard hexadecimal value

## Usage

```
convert_stellarHash(data)
```

## Arguments

| | |
|---|---|
| `data` | base64 encoded hash |

## Value

hex hexadecimal hash

## Examples

```
convert_stellarHash("KMVvhSYRAquk3lPpzljU4SytQSawsTz1aeB+PoKFaf0=")
```

`create_hashFile` *Create sha256 hash of a file*

## Description

Create sha256 hash of a file

## Usage

```
create_hashFile(path)
```

## Arguments

| | |
|---|---|
| `path` | filename (and path, if outside working directory) of a file |

## Value

hash

## Examples

```
create_hashFile("test.rds")
```

---

`create_hashObject` *Create sha256 hash of an object/dataset*

---

## Description

Create sha256 hash of an object/dataset

## Usage

```
create_hashObject(data)
```

## Arguments

| | |
|---|---|
| `data` | any dataset or object |

## Value

hash

## Examples

```
create_hashObject(data)
```

---

`create_ttsFile` *Create trusted timestamp of a file*

---

## Description

Create trusted timestamp of a file

## Usage

```
create_ttsFile(path, proxy_ip = NULL, proxy_port = NULL)
```

## Arguments

| | |
|---|---|
| `path` | filename (and path, if outside working directory) |
| `proxy_ip` | if needed, provide proxy ip |
| `proxy_port` | if needed, provide proxy port |

## Value

url

## Examples

```
create_ttsFile("test.rds")
```

---

`create_ttsObject` *Create trusted timestamp of an object/dataset*

---

## Description

Create trusted timestamp of an object/dataset

## Usage

```
create_ttsObject(data, proxy_ip = NULL, proxy_port = NULL)
```

## Arguments

| | |
|---|---|
| `data` | any dataset or object |
| `proxy_ip` | if needed, provide proxy ip |
| `proxy_port` | if needed, provide proxy port |

## Value

url

## Examples

```
create_ttsObject(data)
```

## get_hash *Retrieve hash from STELLAR network*

### Description

Retrieve hash from STELLAR network

### Usage

```
get_hash(url, proxy_ip = NULL, proxy_port = NULL)
```

### Arguments

| | |
|---|---|
| url | url |
| proxy_ip | if needed, provide proxy ip |
| proxy_port | if needed, provide proxy port |

### Value

hash

### Examples

```
get_hash("https://horizon.stellar.org/transactions/ea0ae0etc")
```

## get_timestamp *Retrieve timestamp from STELLAR network*

### Description

Retrieve timestamp from STELLAR network

### Usage

```
get_timestamp(url, proxy_ip = NULL, proxy_port = NULL)
```

### Arguments

| | |
|---|---|
| url | url |
| proxy_ip | if needed, provide proxy ip |
| proxy_port | if needed, provide proxy port |

## Value

GMT GMT-timestamp

## Examples

```
get_timestamp("https://horizon.stellar.org/transactions/ea0ae0etc")
```

---

# get_url_blockchaintransaction

*Get url of the transaction on STELLAR network (stellarchain.io (non-json))*

---

## Description

Get url of the transaction on STELLAR network (stellarchain.io (non-json))

## Usage

```
get_url_blockchaintransaction(url)
```

## Arguments

| | |
|---|---|
| url | url |

## Value

url url of blockchain transaction

## Examples

```
get_url_blockchaintransaction("https://horizon.stellar.org/transactions/ea0ae0etc")
```

validate_hashFile *Validate hash of a file (created on the fly) with hash on STELLAR network p.s. stellar transactions take between 5-7 seconds. If you validate to soon after creating a timestamp, it will fail...*

## Description

Validate hash of a file (created on the fly) with hash on STELLAR network p.s. stellar transactions take between 5-7 seconds. If you validate to soon after creating a timestamp, it will fail...

## Usage

```
validate_hashFile(url, path, proxy_ip = NULL, proxy_port = NULL)
```

## Arguments

| | |
|---|---|
| url | url |
| path | filename (and path, if outside working directory) |
| proxy_ip | if needed, provide proxy ip |
| proxy_port | if needed, provide proxy port |

## Value

res result of validation

## Examples

```
validate_hashFile("https://horizon.stellar.org/transactions/ea0ae0etc", "test.rds")
```

validate_hashObject *Validate hash of an object/dataset (created on the fly) with hash on STELLAR network p.s. stellar transactions take between 5-7 seconds. If you validate to soon after creating a timestamp, it will fail...*

## Description

Validate hash of an object/dataset (created on the fly) with hash on STELLAR network p.s. stellar transactions take between 5-7 seconds. If you validate to soon after creating a timestamp, it will fail...

## Usage

```
validate_hashObject(url, data, proxy_ip = NULL, proxy_port = NULL)
```

## Arguments

| | |
|---|---|
| `url` | url |
| `data` | any dataset or object |
| `proxy_ip` | if needed, provide proxy ip |
| `proxy_port` | if needed, provide proxy port |

## Value

res result of validation

## Examples

```
validate_hashObject("https://horizon.stellar.org/transactions/ea0ae0etc", data)
```

# Index