

# Package ‘base64url’

July 22, 2025

**Type** Package

**Title** Fast and URL-Safe Base64 Encoder and Decoder

**Version** 1.4

**Description** In contrast to RFC3548, the 62nd character (``+') is replaced with ``-', the 63rd character (``/') is replaced with ``\_". Furthermore, the encoder does not fill the string with trailing ``=" . The resulting encoded strings comply to the regular expression pattern ``[A-Za-z0-9\_-]" and thus are safe to use in URLs or for file names.  
The package also comes with a simple base32 encoder/decoder suited for case insensitive file systems.

**URL** <https://github.com/mlg/base64url>

**BugReports** <https://github.com/mlg/base64url/issues>

**NeedsCompilation** yes

**License** GPL-3

**Encoding** UTF-8

**Imports** backports (>= 1.1.0)

**Suggests** base64enc, checkmate, knitr, microbenchmark, openssl, rmarkdown, testthat

**RoxygenNote** 6.0.1

**VignetteBuilder** knitr

**Author** Michel Lang [cre, aut] (ORCID: <<https://orcid.org/0000-0001-9754-0393>>),  
Apache Foundation [ctb, cph],  
Free Software Foundation [ctb, cph]

**Maintainer** Michel Lang <michellang@gmail.com>

**Repository** CRAN

**Date/Publication** 2018-05-14 09:58:28 UTC

## Contents

|                            |   |
|----------------------------|---|
| base32_encode . . . . .    | 2 |
| base64_urlencode . . . . . | 3 |

**Index****4**

---

base32\_encode*Encode to base32 or Decode from base32*

---

**Description**

Simple RFC4648 base32 encoder/decoder. Pads with “=”.

**Usage**

```
base32_encode(x, use.padding = FALSE)
```

```
base32_decode(x, use.padding = FALSE)
```

**Arguments**

|             |   |
|-------------|---|
| x           | [character(1)]<br>Character vector to encode or decode.   |
| use.padding | [logical(1)]<br>If TRUE, base32_encode returns a string whose length is a multiple of 8, padded with trailing “=” if required. base32_decode expects such a string unless this is set to FALSE (default). The internal algorithm currently works with padding, thus it is faster to set this to TRUE. |

**Value**

[character] of the same length as input x.

**References**

Implementation based on base32 encoder/decoder in the GNU lib: <https://www.gnu.org/software/gnulib/>.

**Examples**

```
x = "plain text"
encoded = base32_encode(x)
decoded = base32_decode(encoded)
print(encoded)
print(decoded)
```

---

|                  |   |
|------------------|---|
| base64_urlencode | <i>Encode to base64 or Decode from base64</i> |
|------------------|---|

---

**Description**

In contrast to RFC3548, the 62nd character ('+') is replaced with '-', the 63rd character ('/') is replaced with '\_'. Furthermore, the encoder does not fill the string with trailing '='. The resulting encoded strings comply to the regular expression pattern "[A-Za-z0-9\_-]" and thus are safe to use in URLs or for file names.

**Usage**

```
base64_urlencode(x)
```

```
base64_urldecode(x)
```

**Arguments**

|   |   |
|---|---|
| x | [character(1)]<br>Character vector to encode or decode. |
|---|---|

**Value**

[character] of the same length as input x.

**References**

Implementation based on base64 encoder/decoder in the Apache Portable Runtime (APR): [https://svn.apache.org/repos/asf/apr/apr/trunk/encoding/apr\\_base64.c](https://svn.apache.org/repos/asf/apr/apr/trunk/encoding/apr_base64.c)

**Examples**

```
x = "plain text"
encoded = base64_urlencode(x)
decoded = base64_urldecode(encoded)
print(encoded)
print(decoded)
```

# Index

`base32_decode (base32_encode)`, [2](#)  
`base32_encode`, [2](#)  
`base64_urldecode (base64_urlencode)`, [3](#)  
`base64_urlencode`, [3](#)