Package 'qrencoder'

October 13, 2022

Title Quick Response Code (QR Code) / Matrix Barcode Creator

Version 0.1.0

Maintainer Bob Rudis <bob@rud.is>

Description Quick Response codes (QR codes) are a type of matrix bar code and can be used to authenticate transactions, provide access to multi-factor authentication services and enable general data transfer in an image. QR codes use four standardized encoding modes (numeric, alphanumeric, byte/binary, and kanji) to efficiently store data. Matrix barcode generation is performed efficiently in C via the included 'libqrencoder' library created by Kentaro Fukuchi.

Depends R (>= 3.1.0), raster

License GPL-2

LazyData true

Suggests testthat

LinkingTo Rcpp

Imports Rcpp, base64enc, png

URL http://github.com/hrbrmstr/qrencoder

BugReports https://github.com/hrbrmstr/qrencoder/issues

RoxygenNote 5.0.1

NeedsCompilation yes

Author Bob Rudis [aut, cre],
Kentaro Fukuchi [ctb] (libqrencoder)

Repository CRAN

Date/Publication 2016-09-16 01:50:10

R topics documented:

`qrencode` *Return a QR encoded string as a matrix*

Description

Useful if you want to do your own post-processing

Usage

```
qrencode(to_encode)
```

Arguments

`to_encode`	the data to encode

Examples

```
qrencode("http://rud.is/b")
```

`qrencoder` *Quick Response Code (QR Code) / Matrix Barcode Creator*

Description

Quick Response codes (QR codes) are a type of matrix bar code and can be used to authenticate transactions, provide access to multi-factor authentication services and enable general data transfer in an image. QR codes use four standardized encoding modes (numeric, alphanumeric, byte/binary, and kanji) to efficiently store data. Matrix barcode generation is performed efficiently in C via the included 'libqrencoder' library created by Kentaro Fukuchi.

Author(s)

Bob Rudis (bob@rud.is)

qrencode_df *Return a QR encoded string as an x, y, z data.frame*

Description

Useful for ggplot::geom_raster

Usage

```
qrencode_df(to_encode)
```

Arguments

to_encode	the data to encode

Examples

```
head(qrencode_df("http://rud.is/b"))
```

qrencode_png *Return a QR encoded string as a base 64 encoded inline png*

Description

Return a QR encoded string as a base 64 encoded inline png

Usage

```
qrencode_png(to_encode)
```

Arguments

to_encode	the data to encode

Note

`` is prepended to the encoded png

Examples

```
cat(qrencode_png("http://rud.is/b"))
```

qrencode_raster *Return a QR encoded string as a raster object*

Description

Return a QR encoded string as a raster object

Usage

```
qrencode_raster(to_encode)
```

Arguments

to_encode	the data to encode

Examples

```
library(raster)
old_mar <- par()$mar
par(mar=c(0,0,0,0))
image(qrencode_raster("http://rud.is/b"), asp=1, col=c("white", "black"),
                                          axes=FALSE, xlab="", ylab="")
par(mar=old_mar)
```

qrencode_raw *Encodes a string as a QR code*

Description

Encodes a string as a QR coder

Usage

```
qrencode_raw(to_encode, version = 0L, level = 0L, hint = 2L,
  caseinsensitive = 1L)
```

Arguments

to_encode	character string to encode
version	version of the symbol. If 0, the library chooses the minimum version for the given input data.
level	error correction level (0 - 3, lowest to highest)

`hint` tell the library how Japanese Kanji characters should be encoded. If "3", the library assumes that the given string contains Shift-JIS characters and encodes them in Kanji-mode. If "2" is given, all of non-alphanumerical characters will be encoded as is. If you want to embed UTF-8 string, choose this. Other mode will cause EINVAL error.

"0" is "numeric mode", "1" is "alphanumeric mode", "5" is "ECI mode".

`caseinsensitive`
case-sensitive(1) or not(0).

See Also

http://www.qrcode.com/en/about/version.html

Index