Package 'ced'

January 14, 2020

Type Package

Title The Compact Encoding Detector

Description R bindings of the Google Compact Encoding Detection library (<https://github.com/google/compact_enc_det>). The library takes as input a source buffer of raw text bytes and probabilistically determines the most likely encoding for that text. It was designed with accuracy, robustness, small size, and speed in mind.

Version 1.0.1

License GPL-2

Copyright file COPYRIGHTS

URL https://artemklevtsov.gitlab.io/ced, https://gitlab.com/artemklevtsov/ced

BugReports https://gitlab.com/artemklevtsov/ced/issues

Depends R (>= 3.5.0)

Imports Rcpp

Suggests tinytest, curl

LinkingTo Rcpp

Encoding UTF-8

NeedsCompilation yes

ByteCompile yes

RoxygenNote 7.0.2

SystemRequirements C++11, GNU make

Author Artem Klevtsov [aut, cre] (<https://orcid.org/0000-0003-0492-6647>), Philipp Upravitelev [ctb], Google Inc. [cph]

Maintainer Artem Klevtsov <a.a.klevtsov@gmail.com>

Repository CRAN

Date/Publication 2020-01-14 09:20:06 UTC

R topics documented:

---

ced *The Compact Encoding Detector*

Description

R bindings of the Google Compact Encoding Detection library.

Author(s)

Maintainer: Artem Klevtsov <a.a.klevtsov@gmail.com> (ORCID)

Other contributors:

- Philipp Upravitelev <upravitelev@gmail.com> [contributor]
- Google Inc. [copyright holder]

See Also

Useful links:

- https://artemklevtsov.gitlab.io/ced
- https://gitlab.com/artemklevtsov/ced
- Report bugs at https://gitlab.com/artemklevtsov/ced/issues

---

ced_enc_detect *Detect Encoding*

Description

Detect charset encoding of the character or raw vector.

Usage

```
ced_enc_detect(x, enc_hint = NULL, lang_hint = NULL)
```

Arguments

x	Raw or character vector.
enc_hint	Character vector with encoding hint.
lang_hint	Character vector with langauge code hint.

Value

Character vector with suggested encodings.

Examples

```
# detect character vector with ASCII strings
ascii <- "I can eat glass and it doesn't hurt me."
ced_enc_detect(ascii)
ced_enc_detect(charToRaw(ascii))

# detect character vector with UTF-8 strings
utf8 <- "\u4e0b\u5348\u597d"
print(utf8)
ced_enc_detect(utf8)
ced_enc_detect(charToRaw(utf8))

# path to examples
ex_path <- system.file("test.txt", package = "ced")
ex_txt <- read.dcf(ex_path, all = TRUE)

# russian text
print(ex_txt[["France"]])
ced_enc_detect(ex_txt[["Russian"]])
ced_enc_detect(iconv(ex_txt[["Russian"]], "utf8", "ibm866"))
ced_enc_detect(iconv(ex_txt[["Russian"]], "utf8", "windows-1251"))
ced_enc_detect(iconv(ex_txt[["Russian"]], "utf8", "koi8-r"))

# chinese text
print(ex_txt[["Chinese"]])
ced_enc_detect(ex_txt[["Chinese"]])
ced_enc_detect(iconv(ex_txt[["Chinese"]], "utf8", "gb18030"))

# korean text
print(ex_txt[["Korean"]])
ced_enc_detect(ex_txt[["Korean"]])
ced_enc_detect(iconv(ex_txt[["Korean"]], "utf8", "uhc"))
ced_enc_detect(iconv(ex_txt[["Korean"]], "utf8", "iso-2022-kr"))

# japanese text
print(ex_txt[["Japanese"]])
ced_enc_detect(ex_txt[["Japanese"]])
ced_enc_detect(iconv(ex_txt[["Japanese"]], "utf8", "shift_jis"))
ced_enc_detect(iconv(ex_txt[["Japanese"]], "utf8", "iso-2022-jp"))



# detect encoding of the web pages content
if (require("curl")) {
  detect_enc_url <- function(u) ced_enc_detect(curl_fetch_memory(u)$content)
  detect_enc_url("https://www.corriere.it")
  detect_enc_url("https://www.vk.com")
  detect_enc_url("https://www.qq.com")
```

```
  detect_enc_url("https://kakaku.com")
  detect_enc_url("https://etoland.co.kr")
}
```

`ced_version` *Compact Encodig Detector Verion*

Description

Backed library version string.

Usage

```
ced_version()
```

Value

Numeric version of the upstream library.

Index