

Package: dexid (via r-universe)

May 9, 2026

Title Generate Memorable Identifiers Using Pokémon Names

Version 0.0.0.9000

Description Provides a simple way to generate human-friendly, memorable identifiers by combining adjectives with Pokémon names. Inspired by codename-style ID generators, this package uses the Pokédex to create unique and fun labels for objects, files, or projects. Useful for naming containers, tracking experiments, or adding a bit of flair to data pipelines.

License MIT + file LICENSE

Encoding UTF-8

Roxygen list(markdown = TRUE)

RoxygenNote 7.3.2

Depends R (>= 3.5)

Imports stringr

URL <https://github.com/schochastics/dexid>,
<https://schochastics.github.io/dexid/>

BugReports <https://github.com/schochastics/dexid/issues>

Suggests testthat (>= 3.0.0)

Config/testthat/edition 3

Config/pak/sysreqs libicu-dev

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

Date/Publication 2025-07-29 08:24:43 UTC

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

RemoteRef HEAD

RemoteSha 564fca80f67e73127c33914307dad5928ea9cdc5

Contents

dex_sentence	2
dexid	3

dex_sentence	<i>Generate Random sentences of Pokémon using moves</i>
--------------	---

Description

Generate Random sentences of Pokémon using moves

Usage

```
dex_sentence(  
  n,  
  gen = 1:9,  
  case = c("sentence", "snake", "kebab", "camel", "caps", "custom"),  
  ...  
)
```

Arguments

n	An integer. Number of IDs to generate.
gen	vector of integers. The generations of Pokemon to use. Defaults to all generations.
case	Character string. The case style to use. Supported cases are snake, kebab, camel, caps, sentence and custom. If case is set to custom, a delimiter argument must be provided.
...	If case is set to custom, a delimiter argument must be provided.

Value

A character vector of length n containing the generated sentences.

Examples

```
dex_sentence(3)  
dex_sentence(5)
```

dexid *Generate Random IDs Using Adjectives and Pokemon Names*

Description

Creates unique identifiers by combining random adjectives with Pokemon names in different formats.

Usage

```
dexid(  
  n,  
  n_adj = 1,  
  gen = 1:9,  
  case = c("snake", "kebab", "camel", "caps", "sentence", "custom"),  
  add_digits = FALSE,  
  ...  
)
```

Arguments

n	An integer. Number of IDs to generate.
n_adj	An integer. Number of adjectives to use. Defaults to 1.
gen	vector of integers. The generations of Pokemon to use. Defaults to all generations.
case	Character string. The case style to use. Supported cases are snake, kebab, camel, caps, sentence and custom. If case is set to custom, a delimiter argument must be provided.
add_digits	Logical. If TRUE, adds a random digit to the end of each ID. Defaults to FALSE.
...	If case is set to custom, a delimiter argument must be provided.

Value

A character vector of length n containing the generated IDs.

Examples

```
# Generate 5 basic snake_case IDs  
dexid(5)  
  
# Generate 5 IDs with 2 adjectives for more uniqueness  
dexid(5, n_adj = 2)  
  
# Generate 5 kebab Case IDs  
dexid(5, case = "kebab")  
  
# Generate 5 IDs with a custom delimiter
```

```
dexid(5, case = "custom", delimiter = "|")

# Generate 5 IDs with random digits
dexid(5, add_digits = TRUE)

# Generate 5 IDs restrict to specific generation(s)
dexid(5, gen = 1)
```

Index

dex_sentence, 2

dexid, 3