

Package: holideh (via r-universe)

May 14, 2026

Title Working with Canadian Dates
Version 0.1.0.9000
Description Convenience date tools for identifying weekends, business days, and Canadian holidays, including R wrappers for the 'Canada Holidays' API <<https://canada-holidays.ca/>>.
License MIT + file LICENSE
URL <https://adamoshen.github.io/holideh/>,
<https://github.com/adamoshen/holideh/>
BugReports <https://github.com/adamoshen/holideh/issues>
Depends R (>= 4.1.0)
Imports cli, dplyr, htr2, lubridate, magrittr, purrr, rlang, stringr, tibble, tidyr, tidymodels, utf8
Suggests fansi, knitr, rmarkdown
VignetteBuilder knitr
Encoding UTF-8
Roxygen list(markdown = TRUE)
RoxygenNote 7.3.3
Config/pak/sysreqs libicu-dev libssl-dev
Repository <https://adamoshen.r-universe.dev>
Date/Publication 2026-04-14 00:10:10 UTC
RemoteUrl <https://github.com/adamoshen/holideh>
RemoteRef HEAD
RemoteSha 1d49cfd05eda616a27a6e433743292907ce1f269

Contents

count_bizdays	2
get_holidays	3
get_province	4
is_bizday	5
off_day	6

count_bizdays	<i>Count the number of business days between a range</i>
---------------	--

Description

Count the number of business days between a range of dates, inclusively.

Usage

```
count_bizdays(from, to, holidays, weekend = c("Sat", "Sun"))
```

Arguments

from	The beginning of the date range.
to	The end of the date range.
holidays	A vector of dates that are holidays.
weekend	A character vector of three-letter abbreviations of weekday names indicating days that should be considered a weekend. Acceptable values are: "Sun", "Mon", "Tue", "Wed", "Thu", "Fri", "Sat".

Details

Holiday dates can be obtained using [get_holidays\(\)](#), [get_province\(\)](#), or by defining a custom vector of holidays.

Value

A single number.

See Also

[is_bizday\(\)](#), [is_holiday\(\)](#), [is_weekend\(\)](#)

Examples

```
library(lubridate)

winter_holidays <- ymd(c("2025-12-25", "2025-12-26"))

count_bizdays(from = ymd("2025-12-20"), to = ymd("2025-12-31"), holidays = winter_holidays)
```

`get_holidays`*Get holidays from the Canada Holidays API*

Description

Get all holidays for a given year. Best used for obtaining a list of federal holidays.

Usage

```
get_holidays(year = NULL, federal = NULL, optional = NULL)
```

Arguments

<code>year</code>	The year for which holidays should be retrieved, between 2013 and 2038. The default, <code>NULL</code> , is equivalent to the current year.
<code>federal</code>	A boolean indicating whether only federal holidays should be retrieved. The default, <code>NULL</code> , is equivalent to <code>FALSE</code> .
<code>optional</code>	A boolean indicating whether optional (non-legislated) holidays should be retrieved. The default, <code>NULL</code> , is equivalent to <code>FALSE</code> .

Value

A [tibble](#) with columns:

date `<date>` The date when the holiday occurs.

observed_date `<date>` The date when the holiday is observed (celebrated). For example, if Christmas Day falls on a Sunday, it is observed (celebrated) on the preceding Monday.

name_en `<chr>` The name of the holiday, in English.

name_fr `<chr>` The name of the holiday, in French.

federal `<lgl>` Whether the holiday is a federal holiday.

holiday_id `<int>` The id of the holiday.

provinces `<list>` A list of tibbles containing information on the provinces observing the holiday and source links.

See Also

[get_province\(\)](#), [Canada Holidays API](#)

Examples

```
if (interactive()) {  
  get_holidays()  
}
```

`get_province`*Get holidays for a province or territory from the Canada Holidays API*

Description

Get all holidays for a given year, for a province or territory. Best used to obtain holiday information for a specific province or territory.

Usage

```
get_province(province, year = NULL, optional = NULL)
```

Arguments

<code>province</code>	The two letter abbreviation for a province/territory (case-sensitive).
<code>year</code>	The year for which holidays should be retrieved, between 2013 and 2038. The default, NULL, is equivalent to the current year.
<code>optional</code>	A boolean indicating whether optional (non-legislated) holidays should be retrieved. The default, NULL, is equivalent to FALSE.

Details

Province and territory codes:

- Alberta: "AB"
- British Columbia: "BC"
- Manitoba: "MB"
- New Brunswick: "NB"
- Newfoundland and Labrador: "NL"
- Nova Scotia: "NS"
- Northwest Territories: "NT"
- Nunavut: "NU"
- Ontario: "ON"
- Prince Edward Island: "PE"
- Quebec: "QC"
- Saskatchewan: "SK"
- Yukon: "YT"

Value

A [tibble](#) with columns:

date <date> The date when the holiday occurs.

observed_date <date> The date when the holiday is observed (celebrated). For example, if Christmas Day falls on a Sunday, it is observed (celebrated) on the proceeding Monday.

name_en <chr> The name of the holiday, in English.

name_fr <chr> The name of the holiday, in French.

federal <lgl> Whether the holiday is a federal holiday.

holiday_id <int> The id of the holiday.

province_id <chr> The abbreviated province/territory code.

province_name_en <chr> The name of the province/territory, in English.

province_name_fr <chr> The name of the province/territory, in French.

source_info <list> A list containing a link to the information source.

See Also

[get_holidays\(\)](#), [Canada Holidays API](#)

Examples

```
if (interactive()) {
  get_province(province = "ON")
}
```

 is_bizday

Detect business days

Description

In a vector of dates, detect the business days (i.e. exclude holidays and weekends).

Usage

```
is_bizday(x, holidays, weekend = c("Sat", "Sun"))
```

Arguments

x	A vector of dates or date-times. If date-times are supplied, the date component will be extracted.
holidays	A vector of dates that are holidays.
weekend	A character vector of three-letter abbreviations of weekday names indicating days that should be considered a weekend. Acceptable values are: "Sun", "Mon", "Tue", "Wed", "Thu", "Fri", "Sat".

Details

Holiday dates can be obtained using `get_holidays()`, `get_province()`, or by defining a custom vector of holidays.

Value

A logical vector of length equal to `x`.

See Also

`is_holiday()`, `is_weekend()`, `count_bizdays()`

Examples

```
library(lubridate)

dates <- seq.Date(from = ymd("2025-12-20"), to = ymd("2025-12-31"), by = "1 day")
winter_holidays <- ymd(c("2025-12-25", "2025-12-26"))

rlang::set_names(is_bizday(dates, holidays = winter_holidays), dates)
```

off_day

Detect non-business days

Description

In a vector of dates, detect the non-business days (holiday or weekend).

Usage

```
is_holiday(x, holidays)

is_weekend(x, weekend = c("Sat", "Sun"))
```

Arguments

<code>x</code>	A vector of dates or date-times. If date-times are supplied, the date component will be extracted.
<code>holidays</code>	A vector of dates that are holidays.
<code>weekend</code>	A character vector of three-letter abbreviations of weekday names indicating days that should be considered a weekend. Acceptable values are: "Sun", "Mon", "Tue", "Wed", "Thu", "Fri", "Sat".

Details

Holiday dates can be obtained using `get_holidays()`, `get_province()`, or by defining a custom vector of holidays.

Value

A logical vector of length equal to *x*.

See Also

[is_bizday\(\)](#)

Examples

```
library(lubridate)

dates <- seq.Date(from = ymd("2025-12-20"), to = ymd("2025-12-31"), by = "1 day")
winter_holidays <- ymd(c("2025-12-25", "2025-12-26"))

rlang::set_names(is_holiday(dates, holidays = winter_holidays), dates)
rlang::set_names(is_weekend(dates), dates)
```

Index

`count_bizdays`, [2](#)
`count_bizdays()`, [6](#)

`get_holidays`, [3](#)
`get_holidays()`, [2](#), [5](#), [6](#)
`get_province`, [4](#)
`get_province()`, [2](#), [3](#), [6](#)

`is_bizday`, [5](#)
`is_bizday()`, [2](#), [7](#)
`is_holiday (off_day)`, [6](#)
`is_holiday()`, [2](#), [6](#)
`is_weekend (off_day)`, [6](#)
`is_weekend()`, [2](#), [6](#)

`off_day`, [6](#)

`tibble`, [3](#), [5](#)