workingless

Release 0.2.1

Julián Cortés

Dec 03, 2020

CONTENTS

Holidays and working days calculations.

CHAPTER

ONE

INSTALLATION

```
$ pip install workingless
```

CHAPTER

TWO

CONTENTS

2.1 Usage

Workingless has four methods for the countries supported, they are: `get_holidays_from_year`, `is_holiday`, `is_working_day` and `get_next_working_day`.

Example usage:

```
import datetime

from workingless import countries

colombia = countries.COL()
holidays = colombia.get_holidays_from_year(year=2020)
print(type(holidays))  # <class 'generator'>
print(list(holidays))
# result:
# [
#     datetime.date(2020, 1, 1),
#     datetime.date(2020, 1, 6),
#     datetime.date(2020, 3, 23),
#     datetime.date(2020, 4, 5),
#     datetime.date(2020, 4, 9),
#     datetime.date(2020, 4, 10),
#     datetime.date(2020, 4, 12),
#     datetime.date(2020, 5, 1),
#     datetime.date(2020, 5, 25),
#     datetime.date(2020, 6, 15),
#     datetime.date(2020, 6, 22),
#     datetime.date(2020, 6, 29),
#     datetime.date(2020, 7, 20),
#     datetime.date(2020, 8, 7),
#     datetime.date(2020, 8, 17),
#     datetime.date(2020, 10, 12),
#     datetime.date(2020, 11, 2),
#     datetime.date(2020, 11, 16),
#     datetime.date(2020, 12, 8),
#     datetime.date(2020, 12, 25)
# ]
colombia.is_holiday(date=datetime.date(2020, 1, 1))  # True
colombia.is_working_day(datetime.date(2020, 1, 1)  # False
colombia.get_next_working_day(datetime.date(2020, 1, 1)  # datetime.date(2020, 1, 2)
```

2.2 API Reference

2.2.1 Contents

Calculators

Calculators implement specific algorithms for holiday calculations.

Easter Calculator

class workingless.calculators.**EasterCalculator**(*days: int*)

Holidays calculations based in easter date. The holiday could be n days +/- from easter date.

Parameters **days** (*int*) – days of difference from easter date

calculate(*year: int*) → datetime.date

It returns easter date +/- days passed in constructor.

Parameters **year** (*int*) – year for calculate holiday

Returns holiday date

Return type datetime.date

Fixed Calculator

class workingless.calculators.**FixedCalculator**(*month: int*, *day: int*)

This is the simpliest implementation. It returns exactly the same date.

Parameters

- **month** – month
- **day** – day

calculate(*year: int*) → datetime.date

It returns exactly the same date given.

Parameters **year** (*int*) – year for calculate holiday

Returns holiday date

Return type datetime.date

Moving Calculator

class workingless.calculators.**MovingCalculator**(*month*, *day*, *next_day=0*)

Holidays calculation based in moving date if date isn't the specific date. For example: base date is january 6, if that date is not monday, holiday will be next monday.

Parameters

- **month** (*int*) – base month
- **day** (*int*) – base day
- **next_day** (*int*) – next day to move holiday

calculate(*year*)
Calculate holiday from base date

Parameters **year** (*int*) – year for calculate holiday

Returns holiday date

Return type datetime.date

Position Day Calculator

class workingless.calculators.**PositionDayCalculator**(*month: int, day: int, position: int, weekday: int = 0*)

Holidays calculation when holiday is position day of month.

For example:

- First monday of february
- Third monday of march

Parameters

- **month** (*int*) – month
- **day** (*int*) – base day, usually first (1)
- **position** (*int*) – position of the month
- **weekday** (*int*) – day of week

calculate(*year: int*) → datetime.date
Calculate holiday based in position day of month

Parameters **year** (*int*) – year for calculate holiday

Returns holiday date

Return type datetime.date

Every N Years Calculator

class workingless.calculators.**EveryNYearsCalculator**(*month: int, day: int, base_year: int, every: int*)

Holidays calculations based in every years from base year

Parameters

- **month** (*int*) – month if every year is satisfied
- **day** (*int*) – day if every year is satisfied
- **base_year** (*int*) – base year for calculation
- **every** (*int*) – how often

calculate(*year: int*) → Optional[datetime.date]
It return base date if the `year` meets the condition

Parameters **year** (*int*) – year for calculate holiday

Returns if year meets the condition, None otherwise.

Return type datetime.date

Utils

`workingless.utils.`**`get_next_day`**(*date: datetime.date*, *next_day: int = 0*) → datetime.date

Get next day if date is not equal than date

Parameters

- **`date`** (*`datetime.date`*) – from date
- **`next_day`** (*`int`*) – next day. MONDAY is default

Returns

same date if day from `date` is equal to next_date, date with next_day otherwise.

Return type date

CHAPTER

THREE

INDICES AND TABLES

- genindex
- modindex
- search

INDEX