sopn-publish-date

Release 1.0.0

Apr 04, 2019

Contents

Overview

Given the polling day of an election in the UK, when should the Statement of Persons Nominated (SoPN) be published?

When an election is called in the United Kingdom, the voting public must have access to the list of candidates who have been nominated to that post.

These documents are called Statements of Persons Nominated, and must be published a set number of working days ahead of the actual election date. The number varies based on:

- *Type of Post* - Parliamentary, Local, devolved Government, etc.

- *Country* - The United Kingdom has up to four different rules for the same type of election, one for each country.

- *Calendar* - each country has their own unique set of Bank Holidays.

This library encapsulates timetable legislation for elections run in the United Kingdom and its devolved administrations.

1.1 Usage

```python
from sopn_publish_date import StatementPublishDate
from datetime import date

publish_date = StatementPublishDate()

print(publish_date.national_assembly_for_wales(date(2016, 5, 5)))

# datetime.date(2016, 4, 7)
```

1.2 Installation

```
pip install sopn_publish_date
```

1.3 Third-Party Sources

The bank holidays JSON used in this project is provided by gov.uk under the Open Government Licence

Legislation

Below are the types of election supported by this library and their corresponding pieces of legislation.

2.1 United Kingdom

2.2 England

2.3 Wales

2.4 Scotland

2.5 Northern Ireland

Tests

Writing tests for code that represents legislation is a bit fiddly.

Ideally we'd have perfect test cases but as noted in *Overview* there are concrete examples of a candidate rejecting their nomination, pushing back the publishing of the SoPN papers.

This project has two sets of tests:

- Unit tests, with single-specified examples

- Approval tests, with test data sourced from parsing historic SoPNs (provided by Democracy Club, who maintain a database of candidates and elections)

API

4.1 sopn_publish_date

class sopn_publish_date.**StatementPublishDate**

    european_parliament(*poll_date: datetime.date*) → datetime.date
        Calculate the publish date for an election to the European Parliament

        This is set out in The European Parliamentary Elections (Amendment) Regulations 2009

            Parameters poll_date – a datetime representing the date of the poll

            Returns a datetime representing the expected publish date

    for_id(*election_id: str*) → datetime.date
        Calculate the publish date for an election given in uk-election-ids format, or raise an exception if that
        election id is ambiguous (could correspond to elections in multiple countries with different electoral legis-
        lation)

            Parameters election_id – a string representing an election id in uk-election-ids format

            Returns a datetime representing the expected publish date

    greater_london_assembly(*poll_date: datetime.date*) → datetime.date
        Calculate the publish date for an election to the Greater London Assembly

        This is set out in The Greater London Authority Elections (Amendment) Rules 2016

            Parameters poll_date – a datetime representing the date of the poll

            Returns a datetime representing the expected publish date

    local(*poll_date: datetime.date*, *country: sopn_publish_date.calendars.Country*)
        Calculate the publish date for a local election.

        This is set out in:

            • The Local Elections (Principal Areas) (England and Wales) (Amendment) Rules 2014

- The Local Elections (Northern Ireland) Order 2010

- The Scottish Local Government Elections Order 2011

> **Parameters**
>
> - **poll_date** – a datetime representing the date of the poll
>
> - **country** – the country in which the election is being run
>
> **Returns** a datetime representing the expected publish date

mayor(*poll_date: datetime.date*) → datetime.date
   Calculate the publish date for an election to the position of Mayor in England and Wales

   This is set out in The Local Authorities (Mayoral Elections) (England and Wales) (Amendment) Regulations 2014

> **Parameters poll_date** – a datetime representing the date of the poll
>
> **Returns** a datetime representing the expected publish date

national_assembly_for_wales(*poll_date: datetime.date*) → datetime.date
   Calculate the publish date for an election to the National Assembly for Wales

   This is set out in The National Assembly for Wales (Representation of the People) (Amendment) Order 2016

> **Parameters poll_date** – a datetime representing the date of the poll
>
> **Returns** a datetime representing the expected publish date

northern_ireland_assembly(*poll_date: datetime.date*) → datetime.date
   Calculate the publish date for an election to the Northern Ireland Assembly

   This is set out by Schedule 5, Rules 1 and 2 of The Northern Ireland Assembly (Elections) (Amendment) Order 2009

> **Parameters poll_date** – a datetime representing the date of the poll
>
> **Returns** a datetime representing the expected publish date

police_and_crime_commissioner(*poll_date: datetime.date*) → datetime.date
   Calculate the publish date for an election to the position of Police and Crime Commissioner

   This is set out in The Police and Crime Commissioner Elections (Amendment) Order 2014

> **Parameters poll_date** – a datetime representing the date of the poll
>
> **Returns** a datetime representing the expected publish date

scottish_parliament(*poll_date: datetime.date*) → datetime.date
   Calculate the publish date for an election to the Scottish Parliament

   This is set out in The Scottish Parliament (Elections etc.) Order 2015

> **Parameters poll_date** – a datetime representing the date of the poll
>
> **Returns** a datetime representing the expected publish date

uk_parliament(*poll_date: datetime.date*, *country: sopn_publish_date.calendars.Country = <Country.ENGLAND: 1>*)
   Calculate the publish date for an election to the Parliament of the United Kingdom

   This is set out in Electoral Registration and Administration Act 2013

> **Parameters**

> • **poll_date** – a datetime representing the date of the poll
>
> • **country** – the country in which the election is being run
>
> **Returns** a datetime representing the expected publish date

4.2 sopn_publish_date.calendars

class sopn_publish_date.calendars.**BankHolidayCalendar**(*dates*)
    A calendar that honours the standard 5-day week in addition to the input list of dates.

class sopn_publish_date.calendars.**Country**
    The countries of the United Kingdom.

> **ENGLAND = 1**
>
> **NORTHERN_IRELAND = 2**
>
> **SCOTLAND = 3**
>
> **WALES = 4**

class sopn_publish_date.calendars.**UnitedKingdomBankHolidays**
    A representation of the bank holiday calendars in the United Kingdom.

> This class exposes a function for each unique calendar: England & Wales, Northern Ireland, and Scotland.

> **england_and_wales**() → sopn_publish_date.calendars.BankHolidayCalendar
>
> > **Returns** a calendar representation of bank holidays in England and Wales

> **from_country**(*country: sopn_publish_date.calendars.Country*) → sopn_publish_date.calendars.BankHolidayCalendar
>   Return the bank holiday calendar for the input country.
>
> > **Parameters country** – the country to retrieve the calendar for
> >
> > **Returns** the corresponding calendar

> **northern_ireland**() → sopn_publish_date.calendars.BankHolidayCalendar
>
> > **Returns** a calendar representation of bank holidays in Northern Ireland

> **scotland**() → sopn_publish_date.calendars.BankHolidayCalendar
>
> > **Returns** a calendar representation of bank holidays in Scotland

sopn_publish_date.calendars.**as_date**(*timestamp*) → datetime.date
    Transforms a pandas._libs.tslibs.Timestamp into a datetime.date object

> **Parameters timestamp** – a pandas Timestamp object
>
> **Returns** the equivalent python date object

sopn_publish_date.calendars.**holiday_from_datetime**(*name: str*, *original_datetime: datetime.datetime*) → pandas.tseries.holiday.Holiday
    Transforms a named datetime.datetime into a pandas.tseries.holiday.Holiday

> **Parameters**
>
> > • **name** – the name of the holiday
> >
> > • **original_datetime** – a representation of the holiday as a datetime
>
> **Returns** the pandas.tseries.holiday.Holiday representation of the datetime

`sopn_publish_date.calendars.`**`working_days`**(*count: int, calendar: sopn_publish_date.calendars.BankHolidayCalendar*)
→ pandas.tseries.offsets.CustomBusinessDay

A pandas representation of a period with the given number of working days using a specified calendar.

> **Parameters**
>
> > • **`count`** – number of working days
> >
> > • **`calendar`** – calendar representing bank holidays in a specific country
>
> **Returns** a number of days to be used in date arithmetic that honours weekends and bank holidays

4.3 sopn_publish_date.election_ids

exception `sopn_publish_date.election_ids.`**`AmbiguousElectionId`**(*election_id: str*)

> An exception type to represent when an election id (usually a group such as *local.2019-05-02*) can correspond to elections in multiple countries with different legislation governing the publish date of Statements of Persons Nominated.

exception `sopn_publish_date.election_ids.`**`InvalidElectionId`**(*election_id: str*)

> An exception type to represent when an election id does not confirm to DemocracyClub's uk-election-ids format

`sopn_publish_date.election_ids.`**`type_and_poll_date`**(*election_id: str) -> (<class 'str'>, <class 'datetime.date'>*)

> Extract election_type (e.g. parl, local, mayor) and poll_date from an election id.
>
> **Parameters** **`election_id`** – a string representing an election id in uk-election-ids format
>
> **Returns** a tuple (election_type: str, poll_date: date)

Python Module Index

s

Index