

Exercise: Holidays from a Web Service

Table of Contents

Background	1
Instructions	2
Import statements	2
<code>get_holidays()</code> function	2
<code>parse_args()</code> function	3
<code>if __name__ == "__main__":</code> statement	3
Using your program	3

Background

Web APIs let you request information using an HTTP GET request and return the requested information in some structured format, such as JSON. For this exercise, we'll use an API that gives us a list of holidays in a country. We'll need to supply a two-letter country code (see the "Alpha 2" column in the table at https://www.nationsonline.org/oneworld/country_code_list.htm) and a four-digit year.

The API URL we will use for this exercise is <https://date.nager.at/Api/v1/Get/COUNTRYCODE/YEAR>, where you replace COUNTRYCODE with the two-letter country code of the country you are interested in, and YEAR with the four-digit year you are interested in. The API sends you back a JSON response which consists of a list of dictionaries, one dictionary per holiday. Below is an example of a response from the API (only the first two dictionaries of the response are shown):

```
[
  {
    "date": "2021-01-01",
    "localName": "New Year's Day",
    "name": "New Year's Day",
    "countryCode": "US",
    "fixed": false,
    "global": true,
    "counties": null,
    "launchYear": null,
    "type": "Public"
  },
  {
    "date": "2021-01-18",
    "localName": "Martin Luther King, Jr. Day",
    "name": "Martin Luther King, Jr. Day",
    "countryCode": "US",
    "fixed": false,
    "global": true,
    "counties": null,
    "launchYear": null,
    "type": "Public"
  },
  ...
]
```

Instructions

For this exercise you will write a program that takes two command-line arguments, a two-letter country code and a four-digit year, uses the web service described above to retrieve holidays for that country and year, and prints out the date and name of each holiday. You will not write a class for this exercise, just functions.

Import statements

You will need to import the following modules:

- `argparse` (or the `ArgumentParser` class from `argparse`)
- `requests` (or the `get` function from `requests`)
- `sys` (or the variable `argv` from `sys`)

`get_holidays()` function

Write a function called `get_holidays()` that takes two arguments: a country code and a year. Plug these values into the API URL (<https://date.nager.at/Api/v1/Get/COUNTRYCODE/YEAR>) as described in the Background section. Use the `requests` module to make an HTTP GET request. Convert the

response to JSON. For each holiday returned by the API, print the date and the name. Here's example output for Belgium (BE) in 2021:

```
2021-01-01: New Year's Day
2021-04-04: Easter Sunday
2021-04-05: Easter Monday
2021-05-01: Labour Day
2021-05-13: Ascension Day
2021-05-24: Whit Monday
2021-07-21: Belgian National Day
2021-08-15: Assumption Day
2021-11-01: All Saints' Day
2021-11-11: Armistice Day
2021-12-25: Christmas Day
```

`parse_args()` function

Write a function called `parse_args()` that takes one argument: a list of command-line arguments. Create an `ArgumentParser` instance that expects two required arguments, a country code and a year. Use your `ArgumentParser` instance to parse the command-line arguments and return them.

If you need a refresher on `argparse`, see <https://docs.python.org/3/library/argparse.html>.

`if __name__ == "__main__":` statement

Write an `if __name__ == "__main__":` statement that does the following:

- passes `sys.argv[1:]` to your `parse_args()` function and stores the result in a variable
- calls your `get_holidays()` function, passing in the parsed arguments obtained in the previous step (the country code and the year)

Using your program

Your program should take two command-line arguments, a country code and a year. Assuming you named your program `holidays.py`, here's one example of how you might run it (Windows users, replace `python3` with `python`):

```
python3 holidays.py US 2021
```