

Expanding Global Holiday Coverage: Batch 4 Country Integration Google Summer of Code 2025 Proposal

Open World Holidays Framework (Python Software Foundation)



ABSTRACT

This project proposes the addition of 10 new countries to the Open World Holidays Framework as part of **Batch 4** of the ongoing country expansion effort. These include Solomon Islands, Somalia, South Sudan, Sudan, Suriname, Syrian Arab Republic, Tajikistan, Togo, Uganda, and Yemen.

The aim is to accurately implement and validate public holidays for each of these countries using official and verifiable sources, while ensuring conformance with the framework's data structure, naming conventions, and testing requirements.

The project emphasizes **global coverage, data quality, and maintainability** - improving the utility of the library for international developers, localization systems, and calendaring tools.

The work will involve careful research, rule implementation (including religious and movable holidays), unit test coverage across multiple years, and documentation. My prior contribution for Trinidad and Tobago (PR #2402) demonstrates readiness to execute this expansion thoroughly and efficiently.

ABOUT ME

Name: Roshan Pradhan

Email: roshanpradhan2048@gmail.com

Github: <https://github.com/Roshan-1024>

Linkedin: <https://www.linkedin.com/in/roshan-pradhan-95911b341/>

LeetCode: <https://leetcode.com/u/Roshan2048/>

Time Zone: IST (UTC +5:30)

Academic Background:

- I am a first year student from IIIT Kalyani pursuing B.Tech in Computer Science Engineering.
- I started programming in Python when I was in high school and I have experience working with other programming languages like C, C++, JS.
- I found out about Holidays Framework through an Open Source Event named Winter Of Code 4.0 conducted by our college where I did my first open source contribution. I have experience working with Holidays Framework since 4 months and I have a decent understanding of the codebase.

Achievements:

- Secured 3rd place among 5100+ registrations in Winter of Code 4.0
- Practiced more than 200 LeetCode problems
- Completed the Machine Learning Specialization by Andrew Ng.

Motivation:

I am extremely excited about this project as it is a combination of technical challenges and real-world impact. And having contributed to Holidays Framework before I'm motivated to take on a larger and more impactful project.

What do I expect from this project and from my mentors?

From my mentors, I would really appreciate their insights, getting feedback on my approach and advice on writing clean and efficient code.

Other Commitments During GSoC:

During the summer, I have no other obligations. Following the conclusion of my vacation, there will be a relatively light workload after classes resume from the second week of July, allowing me to commit approximately 25-30 hours per week to the project on average. Should any circumstances arise that impede the project's progress, I will promptly inform my mentors and make up for it by increasing my workload. In the event of unforeseen obstacles, I am prepared to allocate additional time to the project in the subsequent weeks.

Post-GSoC commitments:

After GSoC, I will continue contributing to Holidays Framework specifically adding support for holidays for other countries. I also want to stay engaged with the community and help other contributors.

Open and Merged PRs:

(Merged) #2289: Migrate Documentation from Sphinx to MkDocs

(Open) #2402: Add Trinidad and Tobago Holidays

Country Expansion: Batch 4 – Addition of 10 New Countries to the Holidays Framework

Synopsis:

This project proposes the addition of 10 new countries to the Open World Holidays Framework as part of Batch 4 of the broader country expansion initiative. The selected countries are: Solomon Islands, Somalia, South Sudan, Sudan, Suriname, Syrian Arab Republic, Tajikistan, Togo, Uganda, and Yemen. The objective is to research, implement, and validate official public holidays for each of these countries using authoritative and verifiable sources.

Benefits to the Community:

- **Global Coverage:** Expands the geographical scope of the framework, making it more useful for international developers and products with global reach.
- **Improved Localization:** Enhances the framework's value for localization tools, calendar systems, and region-aware applications.
- **Data Quality and Consistency:** Adheres to the framework's standards for naming conventions, data structure, and validation.
- **Sustainability:** Builds on a modular and tested implementation model that can be maintained and expanded easily in the future.

Deliverables:

- Holiday implementations for the 10 listed countries with support for both fixed-date and rule-based (e.g., religious/movable) holidays.
- Use of official government websites or other reliable sources to ensure accuracy.
- Full unit test coverage for each country across multiple years to verify behavior.
- Conformance to framework coding standards, naming patterns, and subdivision conventions where applicable.
- Documentation for each country, including source links and relevant notes.
- Separate PRs per country.

Step-by-Step Implementation Plan :

1. Repository and Metadata Setup

- Increment the total supported countries count in `README.md`.
- Add the new country to the supported countries table with appropriate metadata

2. Country File Creation

- Create a new `<country_name>.py` file inside the `holidays/countries/` directory.
- Define the main country class inheriting from `HolidayBase` as an initial setup.
- Add alias classes for ISO 3166-1 alpha-2 and alpha-3 codes.
- Add class-level docstrings, listing all official sources used to derive holiday data.

3. Registry and Initialization

- Import the newly created class in `holidays/countries/__init__.py`.
- Add the new country to the `registry.py` file for global availability.

4. Holiday Definitions

- Implement fixed-date holidays directly.
- Separate religious holidays (e.g., Hindu, Islamic) into additional internal classes when necessary.
- Ensure correct chronological ordering.

5. Localization (l10n)

- Use the `tr()` function for each holiday name to enable translation.
- Add or update `.po` files for relevant locales in the `locale/` directory.
- Provide appropriate `msgstr` translations for each holiday name.

6. Unit Testing

- Create a new test file in `tests/countries/` named `test_<country_code>.py`.
- Write tests for all holidays added, spanning multiple years.
- Include tests for localization.
- Ensure accurate assertions and coverage of edge cases.

7. Code Quality and Maintenance

- Maintain adherence to coding conventions and file structure.
- Ensure logical ordering, correct attribute usage (`start_year`, etc.), and proper class separation.

- Include comments or documentation about non-obvious implementations or known irregularities.

8. Pull Request Submission

- Submit a clean and organized pull request.
- Clearly list all holidays added, classes used, test coverage, and sources.
- Mention any known edge cases or exceptions in the PR description.
- Ensure green CI/CD pipeline checks before requesting review.

Detailed Timeline:

Community Bonding Period (May 8 – June 1)

- Familiarize deeper with the Holidays Framework structure and standards.
- Review previously merged country PRs to extract reusable patterns.
- Discuss plan and country order with mentors.
- Gather reliable sources and references for all 10 countries.
- Prepare initial boilerplate and testing setup for faster implementation once coding starts.

Coding Period (June 2 – August 25)

Implementation will follow a strict **one-country-per-PR** workflow. I will work sequentially and aim to submit each country's implementation once fully tested and reviewed.

Target: ~1 country/week.

Week	Dates	Task
1	June 2 – June 8	Implement & submit PR for Solomon Islands
2	June 9 – June 15	Implement & submit PR for Somalia
3	June 16 – June 22	Implement & submit PR for South Sudan
4	June 23 – June 29	Implement & submit PR for Sudan
5	June 30 – July 6	Implement & submit PR for Suriname
6	July 7 – July 13	Implement & submit PR for Syrian Arab Republic

Midterm Evaluations (July 14 – July 18)

- Ensure at least 5 countries are fully implemented, tested, and merged (or pending review).
- Document work done so far and share progress with mentors.
- Apply feedback and refine earlier PRs if necessary.

Post-Midterm Work (July 14 – August 25)

Continue implementation of remaining countries.

Week	Dates	Task
7	July 14 – July 20	Implement & submit PR for Tajikistan
8	July 21 – July 27	Implement & submit PR for Togo
9	July 28 – August 3	Implement & submit PR for Uganda
10	August 4 – August 10	Implement & submit PR for Yemen
11	August 11 – August 17	Buffer Week / Feedback Handling / Cross-verification
12	August 18 – August 25	Final refinements, doc cleanup, ensure all PRs are merge-ready

Final Submission (August 25 – September 1)

- Submit final work product and evaluations.
- Ensure all contributions are merged or pending final approval.