
How Global Prayer Times Calculates What It Shows

Prayer times, high-latitude rules, Qibla direction,
Hijri dates, Islamic occasions, and calendars

Public calculation guide

Version 1.1 – public methodology used by the global bot

github.com/escalopa/prayer-bot

Contents

1 Purpose and limits	2
2 End-to-end calculation pipeline	2
3 Solar geometry	2
3.1 Prayer event definitions	3
4 Supported calculation methods	3
5 Asr and madhab selection	4
6 High-latitude rules	4
7 Corrections, precision, and caching	5
8 Qibla direction	5
9 Gregorian and Hijri dates	5
9.1 Gregorian date to Julian day	6
9.2 Umm al-Qura lookup used by the bot	6
9.3 Arithmetic fallback outside the table	6
10 Islamic occasions	7
11 Rolling calendar	8
12 Why another timetable may differ	8
13 References	8

1 Purpose and limits

This document explains the user-facing calculations and Islamic information shown by the *global* Telegram bot. It covers prayer times, high-latitude adjustments, Qibla direction and distance, Gregorian–Hijri conversion, Islamic occasions, and the rolling calendar. It intentionally does not describe deployment, databases, or other infrastructure.

It is a transparent calculation guide, not a religious ruling. Local mosques and recognized religious authorities may publish schedules that include safety margins, observational decisions, or conventions different from the options available here. Local moon-sighting decisions may also differ from a calculated Hijri date.

The bot calculates prayer times locally. Google Maps is used only when a location is set, to resolve the IANA timezone, country, city label, and place identifier. Google does not supply the prayer times.

2 End-to-end calculation pipeline

For a user at latitude ϕ , longitude λ , and IANA timezone Z , the bot applies the following pipeline:

1. Accept Telegram coordinates and validate $-90 \leq \phi \leq 90$ and $-180 \leq \lambda \leq 180$.
2. Resolve Z and the ISO country code. Persist coordinates rounded to three decimal places:

$$\phi_s = \frac{\text{round}(1000\phi)}{1000}, \quad \lambda_s = \frac{\text{round}(1000\lambda)}{1000}.$$
3. Select a country-based default calculation method. The user may change it later.
4. Convert the requested instant into timezone Z and take that local civil date.
5. Calculate a full year of solar events for (ϕ_s, λ_s, Z) and cache it by year and settings.
6. Select Fajr, sunrise, Dhuhr, Asr, Maghrib, and Isha for the requested local date.
7. Apply the chosen high-latitude fallback when normal twilight is unavailable.
8. Apply per-prayer corrections and round each non-zero result to the nearest minute.
9. Convert the same local Gregorian date to Hijri. The correction changes the Hijri label and occasion matching, but never moves a prayer time.

Timezone conversion uses the IANA database, so daylight-saving changes and historical offset rules come from the resolved timezone rather than from a fixed UTC offset.

3 Solar geometry

The underlying solar engine implements the NREL Solar Position Algorithm. SPA determines topocentric solar position from the instant, observer location, and astronomical time parameters. The prayer library then searches for transit and target solar elevations.

For intuition, if δ is solar declination, H is the local hour angle, and h is solar elevation, the central spherical relationship is

$$\sin h = \sin \phi \sin \delta + \cos \phi \cos \delta \cos H. \quad (1)$$

For a target elevation h_0 , the corresponding magnitude of hour angle is

$$H_0 = \arccos\left(\frac{\sin h_0 - \sin \phi \sin \delta}{\cos \phi \cos \delta}\right). \quad (2)$$

The morning event uses the solution before solar transit and the evening event uses the solution after transit. If the argument of $\arccos$ lies outside $[-1, 1]$, that elevation is not reached on that date; this is the mathematical reason a high-latitude fallback may be necessary.

Equations 1 and 2 summarize the event condition. The runtime does not use a low-order declination approximation: it delegates topocentric position and event solving to the SPA-based library.

3.1 Prayer event definitions

Fajr

The pre-transit instant at which solar elevation is $-\theta_F$, where θ_F is the method's Fajr angle.

Sunrise

The SPA sunrise event. The bot passes elevation as zero, so the library uses its standard sea-level treatment.

Dhuhr

Solar transit, when the Sun crosses the local meridian. The bot adds no built-in delay; a user correction can add one.

Asr

A post-transit target elevation derived from the selected shadow factor; see Section 5.

Maghrib

The SPA sunset event.

Isha

The post-transit instant at solar elevation $-\theta_I$, except for Umm al-Qura, which uses 90 minutes after Maghrib.

4 Supported calculation methods

Only Fajr and Isha differ among these nine method presets. Sunrise, Dhuhr, Asr, and Maghrib retain the definitions above.

Method	Fajr	Isha	Runtime convention
Muslim World League (MWL)	18°	17°	Default outside explicitly mapped countries.
Egyptian General Authority	19.5°	17.5°	Default for Egypt.
Umm al-Qura	18.5°	90 min	Isha is exactly Maghrib plus 90 minutes.
University of Islamic Sciences, Karachi	18°	18°	Default for Pakistan, India, Bangladesh, and Afghanistan.
ISNA	15°	15°	Default for the United States and Canada.
Diyanet	18°	17°	Default for Turkey.
Kemenag	20°	18°	Default for Indonesia.
MUIS	20°	18°	Default for Singapore.
JAKIM	20°	18°	Default for Malaysia.

For an angle-based method, the event conditions are

$$h(t_F) = -\theta_F, \quad t_F < t_{\text{transit}}, \quad h(t_I) = -\theta_I, \quad t_I > t_{\text{transit}}.$$

For Umm al-Qura, the final Isha rule is

$$t_I = t_M + 90 \text{ minutes}, \quad (3)$$

where t_M is Maghrib. This fixed-duration rule is applied after the general high-latitude adapter, so a selected night-fraction rule can adjust Umm al-Qura Fajr but does not replace its fixed Isha duration.

5 Asr and madhab selection

Let $k = 1$ for Shafi'i, Maliki, and Hanbali timing, and $k = 2$ for Hanafi timing. With topocentric solar declination δ for the day, the target Asr elevation is

$$h_A = \text{arccot}(k + \tan |\delta - \phi|), \quad \text{arccot}(x) = \arctan\left(\frac{1}{x}\right). \quad (4)$$

The solar-event solver finds the post-transit instant t_A at which $h(t_A) = h_A$. The $k = 2$ target is lower in the afternoon sky and therefore normally produces a later Asr time than $k = 1$.

6 High-latitude rules

The normal schedule is marked abnormal if sunrise or sunset is missing, or if astronomical dawn/dusk at -18° is missing. A configured adapter is then run. The three adapters exposed by this bot change a date only when Fajr or Isha is missing and both sunrise and Maghrib exist.

Define the daytime and nighttime durations

$$D = t_M - t_R, \quad N = 24 \text{ hours} - D, \quad (5)$$

where t_R is sunrise and t_M is Maghrib. Each rule chooses fractions p_F and p_I and sets

$$t_F = t_R - p_F N, \quad t_I = t_M + p_I N. \quad (6)$$

Rule	p_F	p_I
Angle based	$\theta_F/60$	$\theta_I/60$
Middle of the night	1/2	1/2
One seventh of the night	1/7	1/7

For example, with an eight-hour night and an 18° angle, the angle fraction is $18/60 = 0.3$, so the fallback places Fajr 2 hours 24 minutes before sunrise and Isha 2 hours 24 minutes after Maghrib.

These three rules require an actual sunrise and sunset. During polar day or polar night, when one of those events is absent, they cannot synthesize a complete schedule. The bot should not imply certainty in that case; users should follow an appropriate local authority.

7 Corrections, precision, and caching

Each prayer has a user correction Δ_p in whole minutes, constrained by the UI/API to $-30 \leq \Delta_p \leq 30$. The final value is

$$t_{p,\text{shown}} = \text{round}_{\text{minute}}(t_{p,\text{base}} + \Delta_p). \quad (7)$$

Rounding is to the nearest minute, not always downward. A correction affects only the selected prayer. The Hijri correction is separate and never changes these instants.

Schedules are calculated one local year at a time and cached by rounded coordinates, timezone, method, madhab, high-latitude rule, all six prayer corrections, and year. This cache changes performance only; it does not change the equations.

8 Qibla direction

The Qibla tool uses the user's saved rounded coordinates and the fixed Kaaba coordinates

$$\phi_K = 21.4225^\circ, \quad \lambda_K = 39.8262^\circ.$$

It calculates the initial great-circle bearing: the direction in which the shortest path over the Earth's surface begins. Let the user coordinates be (ϕ, λ) and define $\Delta\lambda = \lambda_K - \lambda$. After converting all angles to radians,

$$\beta = \text{atan2}(\sin \Delta\lambda \cos \phi_K, \cos \phi \sin \phi_K - \sin \phi \cos \phi_K \cos \Delta\lambda). \quad (8)$$

The displayed clockwise bearing from geographic north is

$$B = \left(\frac{180\beta}{\pi} + 360 \right) \bmod 360. \quad (9)$$

Thus north is 0° , east is 90° , south is 180° , and west is 270° .

The displayed surface distance uses the haversine formula with mean Earth radius

$$R = 6371.0088 \text{ km}.$$

$$a = \sin^2\left(\frac{\phi_K - \phi}{2}\right) + \cos \phi \cos \phi_K \sin^2\left(\frac{\Delta\lambda}{2}\right), \quad (10)$$

$$d = 2R \arcsin \sqrt{a}. \quad (11)$$

The numeric value is a geographic great-circle bearing. The optional live arrow uses the phone's absolute orientation sensor. Calibration, nearby metal, magnetic interference, and the difference between magnetic and geographic north can affect the arrow. The numeric bearing remains available if live orientation is unsupported.

9 Gregorian and Hijri dates

The bot receives a civil instant, converts it to the user's local timezone, and uses that local Gregorian year-month-day. Before Hijri conversion, it applies a user moon-sighting correction $a \in \{-2, -1, 0, 1, 2\}$:

$$G' = G + a \text{ civil days}. \quad (12)$$

The prayer schedule remains attached to G . The correction changes the displayed Hijri label and the Gregorian day on which a Hijri occasion matches.

9.1 Gregorian date to Julian day

For a Gregorian date with year Y , month M , day d , and fraction of day f , January and February are treated as months 13 and 14 of the preceding year. Let

$$(Y', M') = \begin{cases} (Y - 1, M + 12), & M \leq 2, \\ (Y, M), & M > 2, \end{cases}$$

and, for Gregorian dates,

$$A = \left\lfloor \frac{Y'}{100} \right\rfloor, \quad B = 2 + \left\lfloor \frac{A}{4} \right\rfloor - A.$$

The library's Julian-day expression is

$$JD = 1720994.5 + \lfloor 365.25Y' \rfloor + \lfloor 30.6001(M' + 1) \rfloor + B + d + f. \quad (13)$$

9.2 Umm al-Qura lookup used by the bot

The preferred conversion is a published table of Umm al-Qura lunation starts. The bundled table covers Gregorian dates from 14 March 1937 through 16 November 2077. The converter evaluates the adjusted date at UTC noon, calculates JD , and defines

$$CJDN = \lfloor JD \rfloor, \quad MCJDN = CJDN - 2400000. \quad (14)$$

Let L_i be the table of month-start MCJDN values and choose the first i for which $L_i > MCJDN$. Then

$$ILN = i + 16260, \quad (15)$$

$$y = \left\lfloor \frac{ILN - 1}{12} \right\rfloor + 1, \quad (16)$$

$$m = ILN - 12 \left\lfloor \frac{ILN - 1}{12} \right\rfloor, \quad (17)$$

$$d = MCJDN - L_{i-1} + 1. \quad (18)$$

The displayed Hijri date is (y, m, d) localized into the user's selected language.

For the inverse Umm al-Qura mapping, which the library supports but the bot UI does not currently call,

$$ILN = m + 12(y - 1), \quad (19)$$

$$i = ILN - 16260, \quad (20)$$

$$MCJDN = d - 1 + L_{i-1}, \quad (21)$$

$$JD = MCJDN + 2400000 - 0.5, \quad (22)$$

after which the standard inverse Julian-day algorithm produces the Gregorian date.

9.3 Arithmetic fallback outside the table

If G' falls outside the Umm al-Qura table, the bot uses the library's default tabular Islamic calendar. It has a 30-year cycle of 10,631 days, with leap years

$$2, 5, 7, 10, 13, 16, 18, 21, 24, 26, 29.$$

Months alternate between 30 and 29 days, starting with 30; month 12 has 30 days in a leap year. A convenient forward expression for Hijri (y, m, d) is

$$JD_H = 1948438.5 + 354(y - 1) + \left\lfloor \frac{3 + 11y}{30} \right\rfloor + 29(m - 1) + \left\lfloor \frac{m}{2} \right\rfloor + d. \quad (23)$$

The inverse counts complete 30-year cycles, then complete years and alternating months from

$$D = \lfloor JD - 1948438.5 \rfloor. \quad (24)$$

This arithmetic calendar is deterministic, but it is not a claim about observed crescent visibility.

10 Islamic occasions

The bot does not store a separate set of Gregorian occasion dates. For each local Gregorian day G , it calculates the corrected Hijri date

$$H(G, a) = \text{Hijri}(G + a) \quad (25)$$

and matches the resulting Hijri month and day against a curated catalog. Consequently, the user's timezone and Hijri correction determine which local Gregorian day displays an occasion.

Hijri date	Category	Occasion	Reference or qualification
10 Muharram	Fasting	Ashura	Sahih Muslim 1162a .
12 Rabi al-Awwal	Commonly observed	Mawlid al-Nabi	Exact historical date and observance differ; Quran 33:56 , Sahih Muslim 1162e .
27 Rajab	Commonly observed	Isra and Mi'raj	Precise date is not established; Quran 17:1 .
15 Sha'ban	Commonly observed	Mid-Sha'ban	Practices and scholarly assessment of specific evidence differ.
1 Ramadan	Major	Beginning of Ramadan	Quran 2:185 .
21 Ramadan	Major	Last ten nights begin	Surah Al-Qadr ; Sahih al-Bukhari 2017 .
1 Shawwal	Major	Eid al-Fitr	Confirm the local date; Quran 2:185 .
1 Dhu al-Hijjah	Major	First ten days begin	Sahih al-Bukhari 969 .
9 Dhu al-Hijjah	Fasting	Day of Arafah	Recommended fast for non-pilgrims; Sahih Muslim 1162a .
10 Dhu al-Hijjah	Major	Eid al-Adha	Confirm the local date; Quran 22:36 .

The categories “major”, “fasting”, and “commonly observed” are deliberate. Commonly observed entries are not presented as universally agreed dates or practices. Ramadan, Eid, and other lunar dates should be confirmed with the relevant local authority. Occasion reminders are disabled until a user explicitly enables a category.

11 Rolling calendar

The private calendar feed is generated independently for each user from the saved location, timezone, calculation method, madhab, high-latitude rule, per-prayer corrections, and Hijri correction. On every fetch it calculates:

- timed prayer events for the current local day and the following 29 days;
- all-day Islamic occasion events whose corrected Hijri dates fall in the same window.

The feed rolls forward when fetched. The bot does not permanently store 30 future days and does not need a separate daily calendar job. Stable opaque event identifiers let a calendar provider update a changed prayer instead of creating a duplicate. The provider controls its own subscription refresh schedule, so a new day or settings change may not appear immediately.

12 What can cause differences from another timetable?

Two correct implementations can differ because of:

- a different twilight method or Asr shadow factor;
- high-latitude policy;
- elevation, refraction, or solar-position implementation;
- timezone or daylight-saving assumptions;
- institutional safety margins and per-prayer corrections;
- local crescent observation versus calculated Umm al-Qura dates;
- device compass calibration, magnetic interference, or north reference;
- final rounding rules.

The bot stores coordinates only to three decimal places and uses zero elevation. At high latitudes, small changes in target angle can produce large clock-time differences because the Sun crosses the target elevation slowly.

13 Calculation references

The following public algorithm references describe the calculation foundations used by the bot:

1. [Go Prayer v1.1.1 prayer-time definitions](#)
2. [Go SAMPA v1.0.0 solar calculations](#)
3. [Go Hijri v1.0.2 calendar conversion](#)
4. I. Reda and A. Andreas.
Solar Position Algorithm for Solar Radiation Applications.
NREL/TP-560-34302, revised 2008. [NREL report](#)

This document is maintained with the open-source bot. Corrections are welcome through GitHub issues or pull requests.