GARMIN INTERNATIONAL

Garmin Connect Developer Program
Health API

Version 1.2.2

CONFIDENTIAL

Contents

1. Revision History

Version	Date	Revisions
1.0	12/01/2020	Initial revision
1.0.1	04/26/2021	Added Sleep Scores information to sleep summaries
1.0.2	08/02/2021	Backfill policy updated with new rate limits for production-level keys
1.0.3	09/23/2021	Added Health Snapshot to Summary Endpoints
1.0.4	10/13/2021	User Metrics summaries updated with the new field 'enhanced'
1.0.5	06/01/2022	Added HRV Summaries to Summary Endpoints
1.0.6	06/13/2022	Revised calorie fields in daily summaries
1.0.7	04/25/2023	Appendix A updated
1.0.8	08/30/2023	Daily, Epoch, Sleep, Stress summaries, and Appendix A were updated to support new Venu3 features.
1.0.9	10/16/2023	Removing reference for the user access token from PING/PUSH notifications examples in preparation to retire this field.
1.0.10	11/28/23	Added definitions for the stress scores negative values.
1.0.11	02/01/2024	vo2MaxCycling added to the User Metrics Summaries
1.0.12	09/03/2024	Added Skin Temperature summary endpoint
1.1.0	12/01/2024	Backfill policy updated
1.1.1	06/02/2025	bodyBatteryChargedValue and bodyBatteryDrainedValue added to Stress summaries,
1.1.2	06/06/25	Backfill policy updated

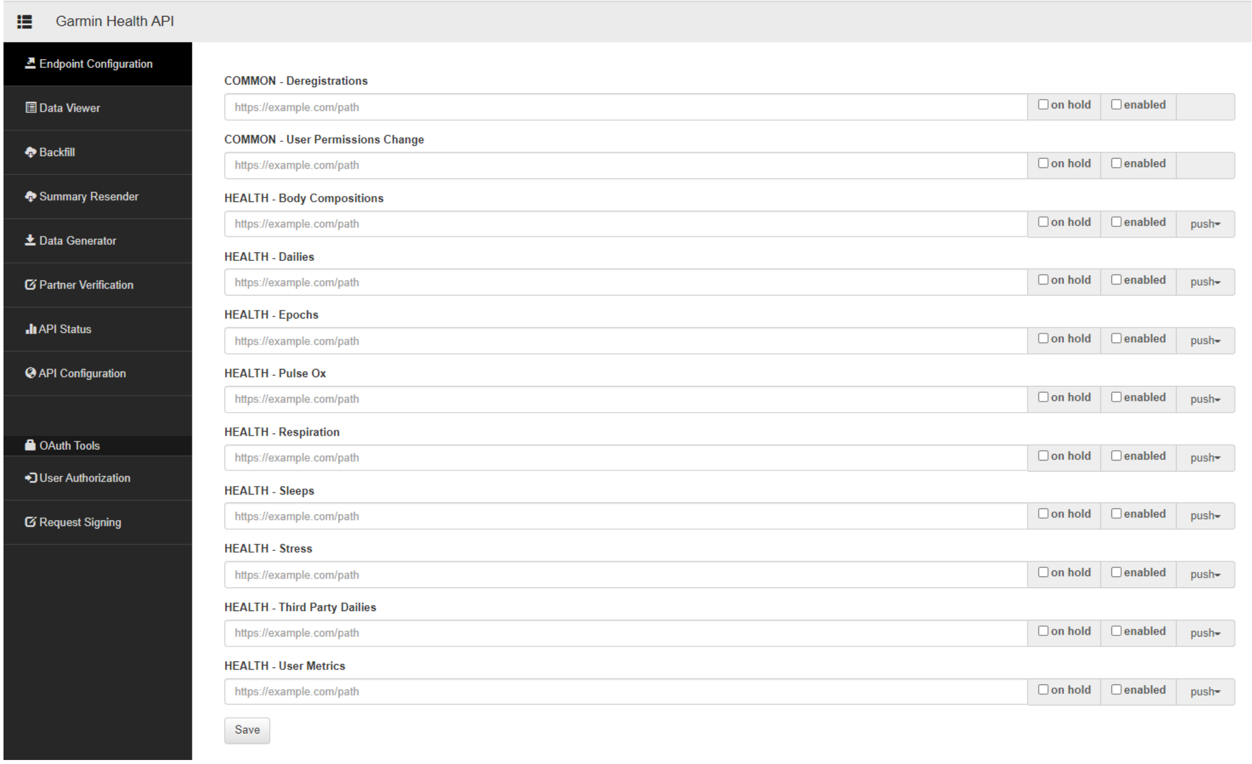
1.2.0	06/30/2025	Production Review requirements updated.
1.2.1	08/14/2025	bodyBatteryChargedValue and bodyBatteryDrainedValue were moved from Stress summaries to Daily summaries
1.2.2	09/09/2025	PING notification updated for OAuth2 clients. Description added for Pull token tool.

2. Purpose of Health API

The Health API lets you leverage valuable health information. After user consent, you can access the all-day health data; everything from detailed sleep level classifications to heart rate and stress. The Health API is ideal for creating integrated corporate wellness, population health, and patient monitoring solutions.

3. Endpoint Configuration

Health API is server-to-server communication only. We deliver event-driven notifications to your configured endpoints. Both the Push Service and the Ping Service can be configured using the Endpoint Configuration Tool found at https://apis.garmin.com/tools/endpoints/. Log in using your consumer key and consumer secret. Below is a screenshot of this tool that shows the configuration possible for each summary type.



COMMON - Deregistrations
https://example.com/path ☐ on hold ☐ enabled

COMMON - User Permissions Change
https://example.com/path ☐ on hold ☐ enabled

HEALTH - Body Compositions
https://example.com/path ☐ on hold ☐ enabled push▾

HEALTH - Dailies
https://example.com/path ☐ on hold ☐ enabled push▾

HEALTH - Epochs
https://example.com/path ☐ on hold ☐ enabled push▾

HEALTH - Pulse Ox
https://example.com/path ☐ on hold ☐ enabled push▾

HEALTH - Respiration
https://example.com/path ☐ on hold ☐ enabled push▾

HEALTH - Sleeps
https://example.com/path ☐ on hold ☐ enabled push▾

HEALTH - Stress
https://example.com/path ☐ on hold ☐ enabled push▾

HEALTH - Third Party Dailies
https://example.com/path ☐ on hold ☐ enabled push▾

HEALTH - User Metrics
https://example.com/path ☐ on hold ☐ enabled push▾

Each enabled summary should be configured with a valid HTTPS URL to which Ping or Push notifications for that summary type will be sent. Other protocols and non-standard ports are not supported. Please make sure the enabled URLs do exist and accept HTTPS POST requests.

Enabled: When checked, this summary data will be made available for all users associated with this consumer key and the summary type will be sent to the provided URL. When unchecked, data will *not* be made available, notifications will not be sent, and any Pings or Pushes in queue (including failed) will be dropped.

On Hold: When checked, data will continue to be available, but notifications will be queued and not sent. Pings and Pushes will be queued for up to seven days and then dropped. When unchecked, all previously queued notifications will be sent serially. If a summary type is not Enabled this setting has no effect.

Tip: On Hold functionality is useful for planned maintenance events or any other instance when it would be useful to temporarily stop the flow of notifications without data loss. Although a missed notification will be re-attempted for as long as possible, using On Hold guarantees seven days of availability as well as resumption of notifications within 2 minutes of disabling the setting. Normal resumption time may be longer due to exponential back-off between failed notification re-attempts.

4. Ping Service (For Ping/Pull Integrations Only)

Garmin will send HTTPS POST ping notifications regarding the availability of new summaries and de-registrations to partners shortly after new data is available. This Ping Service allows partners to maintain near-real-time consistency with the Garmin data store without wasted queries on users who haven't synced any new data.

Access to GCDP APIs is restricted to server-to-server communication; access by end-user devices is not allowed. APIs are designed for a secure one-time transfer of data from Garmin to Partner servers; ad-hoc requests for data are not permitted. Partners are responsible for receiving and storing data in a timely manner. PING notifications are guaranteed 7 days after receipt (Activity Files – 24 hours).

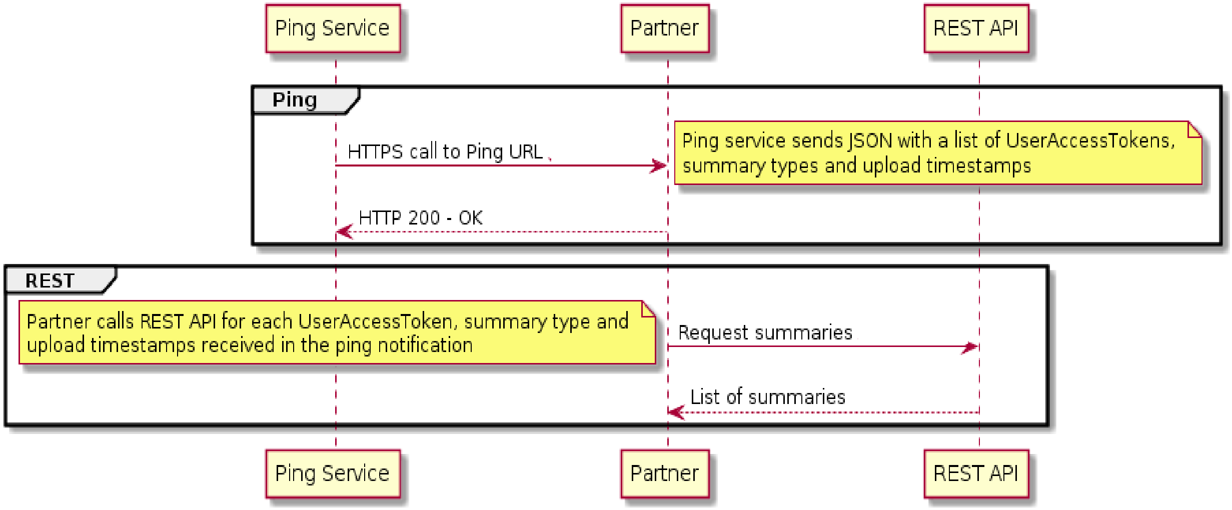
Each notification also contains a callback URL. When this URL is called, data specific to that user and summary type is returned. The partner may provide separate URLs for each summary type for flexible processing or may choose to send ping notifications for all data types to the same endpoint.

Tip: Please call the Health REST API asynchronously after closing the connection of the ping request. One frequent ping/pull implementation mistake is to hold the incoming ping notification HTTP POST open while performing the corresponding the callbacks to the Health API. This will result in HTTP timeouts and potential data loss.

Each ping message contains a JSON structure with a list of userIDs for which new data is available, as well as the URL to call to fetch that data. A successful ping-based integration should never need to call the Health API except as prompted by ping notifications.

4.1.Ping Workflow

The following diagram illustrates the general workflow.


Ping Service
Partner
REST API
Ping
HTTPS call to Ping URL
Ping service sends JSON with a list of UserAccessTokens, summary types and upload timestamps
HTTP 200 - OK
REST
Partner calls REST API for each UserAccessToken, summary type and upload timestamps received in the ping notification
Request summaries
List of summaries
Ping Service
Partner
REST API

The Ping Service has a timeout of thirty seconds. To avoid missed data or improper error responses, it is required to respond to each notification with an HTTP status code of 200 (OK) before performing callbacks to the Health API. Holding the ping open while performing callbacks is the most common cause of instability in Health API integrations.

A failed ping notification is defined as any of the following:

- The partner’s ping endpoint is unreachable
- The endpoint responds with an HTTP status code other than 200
- An error occurs during the request (e.g. the connection breaks)

In the case of a failed ping notification, the Ping Service attempts to re-send the ping regularly. The Ping Service will continue to re-attempt failed pings, successively waiting longer between each attempt, for as long as the failed ping queue depth does not affect the performance of the overall Health API.

Tip: If you know in advance that your notification end points will be unavailable (e.g. server maintenance), you may set your notification to “On Hold” using the Ping Configuration Web Tool (see Web Tools below). Doing so will guarantee quick transmission of pings once the on-hold state is removed and avoid data loss.

In the event of an unexpected outage in which notifications are accepted with HTTP 200s, but the resulting callbacks fail, please contact the Health API Support team (connect-support@developer.garmin.com). They will be happy to help set up a regeneration of all missed notifications during the affected time.

4.2. Ping Notification Content

JSON Element	Description
summary type (list key)	The summary type of this list of pings Valid types: dailies, epochs, sleeps, bodyComps, stressDetails, userMetrics, pulseox, allDayRespiration, healthSnapshot, hrv, bloodPressures, skinTemp
userId	A unique user identifier corresponding to the underlying Garmin account of the user. This userId is *not* used as a parameter for any call to the Health API.
callbackURL	Pre-formed URL to pull the data. Not present for deregistration notifications.

Example

```
{
     “epochs”: [{
          “userId”: “4aacafe82427c251df9c9592d0c06768”,
          “callbackURL”: “https://apis.garmin.com/wellness-
api/rest/epochs?uploadStartTimeInSeconds=1444937651&uploadEndTimeInSe
conds=1444937902&token=XXXXXXXXX”
     }, {}]
}
```

Tip: During your Ping Service integration development, it may be cumbersome for your endpoints to be publicly available to receive real notifications from the Health API. Simulating ping requests within the local network by using tools like cURL is a useful way to solve this problem.
You can generate a temporary pull token via API Pull Token Tool.

Here is an example for simulating a ping request for epoch summaries for a service running on localhost, port 8080:

```
curl -v -X POST "http://localhost:8080/garmin/ping" \
-H "Content-Type: application/json;charset=UTF-8" \
-H "user-agent: Garmin Health API" \
-H "accept: text/plain, application/json, application/*+json, */*" \
-H "accept-encoding: gzip, x-gzip, deflate" \
-H "garmin-client-id: <CLIENT_ID>" \
-H "x-forwarded-for: 204.77.163.244" \
-H "x-forwarded-proto: https" \
-d '{
  "epochs": [
    {
      "userId": "<USER_ID>",
      "callbackURL": "https://apis.garmin.com/wellness-
api/rest/epochs?uploadStartTimeInSeconds=<START_TIME>&uploadEndTimeIn
Seconds=<END_TIME>&token=<TOKEN>"
    }
  ]
}'
```

5. Push Service

Like the Ping Service, the Push Service allows partners to receive near-real-time updates of Garmin user data without delay or duplication associated with regularly scheduled update jobs. Unlike the Ping Service's callback URLs, the Push Service generates HTTPS POSTs that contain the updated data directly within the POST as JSON. This data is the exact same data that the Health API would have returned had a Ping notification been generated and its callback URL invoked; it is purely a matter of preference and ease of integration whether to use the Ping or Push Service.

> **Note:** Push notifications have the same retry logic using the same definition of a failed notification as the Ping Service and support the same On Hold functionality as the Ping service.

5.1.Push Notification Content

JSON Element	Description
summary type (list key)	The summary type of this list of pings. Valid types: dailies, epochs, sleeps, bodyComps, stressDetails, userMetrics, pulseox, allDayRespiration, healthSnapshot, hrv, bloodPressures, skinTemp
userId	A unique user identifier corresponding to the underlying Garmin account of the user. This userId is *not* used as a parameter for any call to the Health API.
Summary data	The summary data in the same data model as the Health API. See the Summary Endpoints section for details and examples of each summary data model.

Example

```
{

   “epochs”: [
      {
         “userId”: “4aacafe82427c251df9c9592d0c06768”,
         “summaryId”: “x153a9f3-5a9478d4-6”,
         “activityType”: “WALKING”,
         “activeKilocalories”: 24,
         “steps”: 93,
         “distanceInMeters”: 49.11,
         “durationInSeconds”: 840,
         “activeTimeInSeconds”: 449,
         “startTimeInSeconds “: 1519679700,
         “startTimeOffsetInSeconds”: -21600,
         “met”: 3.3020337,
         “intensity”: “ACTIVE”,
         “meanMotionIntensity”: 4,
         “maxMotionIntensity”: 7
      }
   ]
}
```

6. Health API Integration Tips

This section describes functionality that is important to understand when integrating with the Garmin Connect Health API and tools to help accelerate and verify that integration.

6.1.Updated Summary Records

The Health API provides updates to previously issued summary records. Updates are summary data records for a given user with the same start time and summary type as a previous summary data record and a duration that is either equal to or greater than the previous summary data’s duration. Updates indicate that newer and possibly more complete data is available for the time period associated with that summary. Garmin Connect users may sync their data multiple times throughout the day, sometimes from multiple devices. Each sync may generate updates and the latest summary should always take precedence over previous records.

Updated summary records may also occur if the user syncs data from multiple devices that have recorded data across the same time period. Garmin Connect automatically merges data from multiple devices, choosing the data most advantageous (e.g. highest step count) to the user.

Important: Your integration should replace old records with the updated summary information. Discarding updates will result in inaccurate information for your program and a data mismatch between Garmin Connect and your platform.

Daily Summary Example: When a user syncs data throughout the day, the summary for that day will be updated.

Epoch Summary Example: If a user syncs 12 minutes into an epoch (i.e. an epoch with durationInSeconds = 720), their next sync (assuming it is at least 3 minutes later) would contain all the data from that specific time period (i.e. durationInSeconds = 900 with the same start time). This newer, complete data should replace the old epoch data.

Multiple Devices Example: If a user goes for a run, they might wear one device to the park and then switch to a different device to record their run. When the user syncs Device 1, it might result in an Epoch summary with only 80 steps but a full 900 duration. If they then sync Device 2, that data might indicate 3,000 steps for the same time period and the same 900 duration. Garmin will automatically merge these two data feeds into a single reconciled Epoch record, which will then be displayed to the user through Garmin Connect. If the updated Epoch record is different than the original Epoch record sent via the Health API a new Ping or Push will be generated and the updated Epoch data should replace the old data, even though the durations are both 900.

6.2.Time Values in the Health API

All timestamps in the Health API are UTC in seconds, also known as Unix Time. However, summary data records may also contain a time offset value. This value represents the difference between the standardized UTC timestamp and the time that displayed on the user's device when the data was generated, or on the designated primary activity tracker for users with multiple devices.

Note that this is not the same as an international standard time zone offset. While devices with GPS offer to set the time automatically and Garmin Connect Mobile can set device time based on the smartphone, users may manually override the time using the settings on the device. Users may change the display time to anything they wish within 24 hours of UTC.

Health API integrations should accommodate the fact that users are given the flexibility to set non-standard display times by either working entirely in UTC, trusting the user's presentation of time, or maintaining a preferred standard time zone separate from and outside of the Health API. For ease of use, summary data types that are one-per-day (such as Dailies) also contain a 'calendarDate', a date stamp corresponding to the user's day with which that record will be associated and displayed in Garmin Connect systems with no time zone manipulation required.

6.3.Web Tools

Several web-based tools are available to assist partners with Health API integration in addition to the Endpoint Configuration tool. These tools are all available by logging in to https://apis.garmin.com/tools/login using the consumer key and secret applicable to the program they want to configure.

6.3.1. Data Viewer

The Data Viewer tool allows viewing of a user's Health API data by summary start and end time to debug or assist an end user. This is the same data that can be pulled from the Health API but allows for additional query options and easier interpretation.

6.3.2. Backfill

The Backfill tool provides a web-based method to initiate historic data requests as described in the Summary Backfill section without the need to access the API programmatically.
Backfill is limited to 1 request per connected user per timeframe. To request Backfill once more, please reconnect to the app.

6.3.3. Summary Resender

The Summary Resender tool regenerates and re-sends all notifications for the provided UATs for the configured summary types. This tool is useful for integration testing and for recovering from outages where Ping or Push notifications were accepted with HTTP 200s, but summary data was not successfully retrieved or stored.

Even so, the use of this tool would be tedious in the event of a system-wide outage. The Garmin Connect Developer support team (connect-support@developer.garmin.com) is happy to help regenerate notifications for all users of a given consumer key for all summary types.

6.3.4. Data Generator

The Data Generator simulates a user syncing data from their device. Semi-randomized data is uploaded to the Health API per provided UAT and notifications are generated for this simulated data. This provides a quick way to test summary data integration changes without needing to actually generate the data on a Garmin device repeatedly.

Please note that to request a production-level key (see Requesting a Production Key above), data synced from actual devices is required.

6.3.5. Partner Verification

As described in the Getting Started section, the Partner Verification tool quickly checks for all requirements in order to be granted access to a Production key.

Tip: Before requesting a production key, please make sure your integration meets these basic requirements:

- Summary data endpoints should only be called as a result of Ping notifications, and only in accordance with the Ping callback URL.
- Push notifications, if configured, must be responded to with an HTTP status code 200 in a timely manner.
- Integrations must have queried or received data from at least two different Garmin Connect accounts where data was uploaded recently by physical Garmin devices.
- Deregistration endpoint enabled, and tested

6.3.6. Pull Token tool

This tool generated a temporary pull token that applications can use to securely access users' data through authorized, time-bound API calls.
The token generated must be passed as a parameter (e.g., ?token=XXXXXX) and an API call to pull users' data.
Once expires, the new token must be generated. Expired tokens cannot be reused.

7. Summary Endpoints

This section provides details of the data available for each summary type. Summary data records are the core method of data transfer in the Health API, with each summary corresponding to a different ping notification type.

PING notifications: All summary data endpoints have a maximum query range of 24 hours **by upload time**. The upload time corresponds to when the user synced the data, not the timestamps of the summary data itself. Since users may have multiple devices that record data from overlapping time periods and they may sync these devices sporadically, querying by upload time prevents needing to infinitely re-query previous time spans to catch new data.

For example, if a user syncs 13 days of data from their device on 11/10/2017 (starting at 18:00:09 and finishing at 18:00:11 GMT), the resulting ping notification would have a start time of 1510336809 and an end time of 1510336811. A call to retrieve the Daily summaries for that range will return all 13 Daily Summaries. This query-by-upload-time mechanism removes any need to query arbitrary lengths into the past just in case the user waits longer than expected between device syncs.

Summary data obtained through **Push** notifications follow the same data model described in this section with the addition of the userAccessToken as described in the Push Service section above.

7.1.Daily Summaries

Daily summaries offer a high-level view of the user's entire day. They generally correspond to the data found on the "My Day" section of Garmin Connect. Daily summaries are the most commonly used and are often the foundation of a Health API integration.

Each daily summary may contain the following fields:

Property	Type	Description
summaryId	string	Unique identifier for the summary.
calendarDate	string	The calendar date of this summary will be displayed on Garmin Connect. The date format is 'yyyy-mm-dd'.
startTimeInSeconds	integer	Start time of the activity in seconds since January 1, 1970, 00:00:00 UTC (Unix timestamp).
startTimeOffsetInSeconds	integer	Offset in seconds to add to startTimeInSeconds to derive the "local" time of the device that captured the data.
activityType	string	This field is included in daily summaries for backward compatibility purposes. It can be ignored and will always default to 'GENERIC'.
durationInSeconds	integer	Length of the monitoring period in seconds. 86400 once a full day is complete, but less if a user syncs mid-day.
steps	integer	Count of steps recorded during the monitoring

		period.
pushes	integer	Count of pushes recorded during the monitoring period. * This field will be present only if the user’s device is in wheelchair mode.
distanceInMeters	floating point	Distance traveled in meters.
pushDistanceInMeters	floating point	Distance traveled in meters. * This field will be present only if the user’s device is in wheelchair mode
activeTimeInSeconds	integer	The portion of the monitoring period (in seconds) in which the device wearer was considered Active. This relies on heuristics internal to each device.
activeKilocalories	integer	Active kilocalories (dietary calories) burned during the monitoring period. This includes only the calories burned by the activity and not calories burned as part of the basal metabolic rate (BMR).
bmrKilocalories	integer	BMR Kilocalories burned by existing Basal Metabolic Rate (calculated based on user height/weight/age/other demographic data).
moderateIntensityDurationInSeconds	integer	Cumulative duration of activities of moderate intensity. Moderate intensity is defined as activity with MET value range 3-6.
vigorousIntensityDurationInSeconds	integer	Cumulative duration of activities of vigorous intensity. Vigorous intensity is defined as activity with MET value > 6.
floorsClimbed	integer	Number of floors climbed during the monitoring period.
minHeartRateInBeatsPerMinute	integer	Minimum of heart rate values captured during the monitoring period, in beats per minute.
averageHeartRateInBeatsPerMinute	integer	Average of heart rate values captured during the last 7 days, in beats per minute. The average heart rate value for the monitoring period can be calculated based on the data from timeOffsetHeartRateSamples.
maxHeartRateInBeatsPerMinute	integer	Maximum heart rate values were captured during the monitoring period, in beats per minute.
restingHeartRateInBeatsPerMinute	Integer	Average heart rate at rest during the monitoring period, in beats per minute.
timeOffsetHeartRateSamples	Map	Collection of mappings between offset from start time (in seconds) to a heart rate value recorded for that time, in beats per minute. Each entry is a representative sample of the previous 15 seconds from the given offset. Lack of entry for a given offset should be interpreted as no data available. For example, in the response below, the user had 75 BPM for the first 30 seconds of the daily summary, took off their device until the 3180 second time slice, and took it off again after the 3255 second entry.

averageStressLevel	integer	An abstraction of the user’s average stress level in this monitoring period, measured from 1 to 100, or -1 if there is not enough data to calculate average stress. Scores between 1 and 25 are considered “rest” (i.e not stressful), 26-50 as “low” stress, 51-75 “medium” stress, and 76-100 as “high” stress.
maxStressLevel	integer	The highest stress level measurement taken during this monitoring period.
stressDurationInSeconds	integer	The number of seconds in this monitoring period where stress level measurements were in the stressful range (26-100).
restStressDurationInSeconds	integer	The number of seconds in this monitoring period where stress level measurements were in the restful range (1 to 25).
activityStressDurationInSeconds	integer	The number of seconds in this monitoring period where the user was engaging in physical activity and so stress measurement was unreliable. All duration in this monitoring period not covered by stress, rest, and activity stress should be considered Uncategorized, either because the device was not worn or because not enough data could be taken to generate a stress score.
lowStressDurationInSeconds	integer	The portion of the user’s stress duration where the measured stress score was in the low range (26-50).
mediumStressDurationInSeconds	integer	The portion of the user’s stress duration where the measured stress score was in the medium range (51-75).
highStressDurationInSeconds	integer	The portion of the user’s stress duration where the measured stress score was in the high range (76-100).
stressQualifier	string	A qualitative label was applied based on all stress measurements in this monitoring period. Possible values: unknown, calm, balanced, stressful, very_stressful, calm_awake, balanced_awake, stressful_awake, very_stressful_awake. This matches what the user will see in Garmin Connect. It is recommended that implementations that use the stressQualifier be tolerant of unknown values in case more granular values are added.
bodyBatteryChargedValue	integer	The amount by which the Body Battery level increased during the monitoring period
bodyBatteryDrainedValue	integer	The amount by which the Body Battery level decreased during the monitoring period
stepsGoal	integer	The user’s steps goal for this monitoring period.
pushesGoal	integer	The user’s pushes goal for this monitoring period. * This field will be present only if the user’s device is in wheelchair mode

intensityDurationGoalInSeconds	integer	The user’s goal for consecutive seconds of moderate to vigorous intensity activity for this monitoring period.
floorsClimbedGoal	integer	The user’s goal for floors climbed in this monitoring period.

Example

```
[
        {
                “summaryId”: “ EXAMPLE_67891”,
                “calendarDate”: “2016-01-11”,
                “activityType”: “WALKING”,
                “activeKilocalories”: 321,
                “bmrKilocalories”: 1731,
                “steps”: 4210,
                “pushes”:10,
                “distanceInMeters”: 3146.5,
                “pushDistanceInMeters”: 32,5,
                “durationInSeconds”: 86400,
                “activeTimeInSeconds”: 12240,
                “startTimeInSeconds”: 1452470400,
                “startTimeOffsetInSeconds”: 3600,
                “moderateIntensityDurationInSeconds”: 81870,
                “vigorousIntensityDurationInSeconds”: 4530,
                “floorsClimbed”: 8,
                “minHeartRateInBeatsPerMinute”: 59,
                “averageHeartRateInBeatsPerMinute”: 64,
                “maxHeartRateInBeatsPerMinute”: 112,
                “timeOffsetHeartRateSamples”: {
                        “15”: 75”,
                        “30”: 75,
                        “3180”: 76,
                        “3195”: 65,
                        “3210”: 65,
                        “3225”: 73,
                        “3240”: 74,
                        “3255”: 74
                },
                “averageStressLevel”: 43,
                “maxStressLevel”: 87,
                “stressDurationInSeconds”: 13620,
                “restStressDurationInSeconds”: 7600,
                “activityStressDurationInSeconds”: 3450,
                “lowStressDurationInSeconds”: 6700,
                “mediumStressDurationInSeconds”: 4350,
                “highStressDurationInSeconds”: 108000,
                “stressQualifier”: “stressful_awake”,
                “stepsGoal”: 4500,
                “pushesGoal”: 100,
        “intensityDurationGoalInSeconds”: 1500,
        “floorsClimbedGoal”: 18
    },
    {
        “summaryId”: “ EXAMPLE_67892”,
        “activityType”: “WALKING”,
        “activeKilocalories”: 304,
        “bmrKilocalories”: 1225,
        “steps”: 3305,
        “pushes”:10,
        “distanceInMeters”: 2470.1,
        “pushDistanceInMeters”: 32,5,
        “durationInSeconds”: 86400,
        “activeTimeInSeconds”: 7,
        “startTimeInSeconds”: 1452556800,
        “startTimeOffsetInSeconds”: 3600,
        “moderateIntensityDurationInSeconds”: 83160,
        “vigorousIntensityDurationInSeconds”: 3240,
        “floorsClimbed”: 5,
        “minHeartRateInBeatsPerMinute”: 62,
        “averageHeartRateInBeatsPerMinute”: 67,
        “maxHeartRateInBeatsPerMinute”: 122,
        “restingHeartRateInBeatsPerMinute”: 64,
        “timeOffsetHeartRateSamples”: {
            “15”: 77”30”: 72,
            “3180”: 71,
            “3195”: 67,
            “3210”: 62,
            “3225”: 65,
            “3240”: 71,
            “3255”: 81
        },
        “averageStressLevel”: 37,
        “maxStressLevel”: 95,
        “stressDurationInSeconds”: 19080,
        “restStressDurationInSeconds”: 2700,
        “activityStressDurationInSeconds”: 7260,
        “lowStressDurationInSeconds”: 7800,
        “mediumStressDurationInSeconds”: 8280,
        “highStressDurationInSeconds”: 3000,
        “stressQualifier”: “stressful_awake”,
        “stepsGoal”: 5000,
        ”pushesGoal”: 100,
        “intensityDurationGoalInSeconds”: 1800,
        “floorsClimbedGoal”: 20
    }
]
```

7.2.Epoch Summaries

This service provides the ability to retrieve a list of summaries containing wellness data for a specific time range. Epoch summary records contain much of the same data available in Daily summaries, but with 15-minute time-slice granularity.

There is one record for each activity type monitored within an individual epoch. For example, if the user was sedentary for five minutes, walked for five minutes, and then ran for five minutes over the course of 15 minutes, three activity records would be generated for that single 15-minute epoch. The duration value would be 900 seconds for all three records, but the active time for each would be 300 seconds.

A duration of less than 900 seconds indicates that the user synced data during the middle of an epoch. On the user's next sync, that epoch record will be replaced with a 900-second-duration epoch covering the entire span. As such and to accommodate users with multiple devices, it is important that new epochs always replace existing epochs that have the same startTimeInSeconds. The most recent update from the Health API will always reflect the most recent data in Garmin Connect.

Epoch data is useful when attempting to construct charts showing intraday wellness data. An example of this in Garmin Connect is the Steps Details chart that graphs step count changes throughout the user's day.

Each wellness monitoring summary may contain the following parameters:

Property	Type	Description
summaryId	string	Unique identifier for the summary.
startTimeInSeconds	integer	Start time of the monitoring period in seconds since January 1, 1970, 00:00:00 UTC (Unix timestamp).
startTimeOffsetInSeconds	integer	Offset in seconds to add to startTimeInSeconds to derive the "local" time of the device that captured the data
activityType	string	Text description of the activity type. See Appendix A for a complete list.
durationInSeconds	integer	Length of the monitoring period in seconds.
activeTimeInSeconds	integer	Portion of the monitoring period (in seconds) in which the device wearer was active for this activity type. The sum of active times of all epochs of the same start time (and different activity types) should be equal to the duration.
Steps	integer	Count of steps recorded during the monitoring period
pushes	integer	Count of pushes recorded during the monitoring period. * This field will be present only if the user's device is in wheelchair mode
distanceInMeters	floating point	Distance traveled in meters
pushDistanceInMeters	floating point	Distance traveled in meters. * This field will be present only if the user's device is in wheelchair mode

activeKilocalories	integer	Active kilocalories (dietary calories) burned during the monitoring period. This includes only the calories burned by the activity and not calories burned as part of the basal metabolic rate (BMR).
Met	floating point	MET (Metabolic Equivalent of Task) value for the active time for this activity type. See Appendix C.
intensity	string	Qualitative measure of intensity. See Appendix B.
meanMotionIntensity	floating point	The average of motion intensity scores for all minutes in this monitoring period. See Appendix D for information on motion intensity.
maxMotionIntensity	floating point	The largest motion intensity score of any minute in this monitoring period. See Appendix D for information on motion intensity.

Example

```
[
        {
                “summaryId”: “EXAMPLE_1234”,
                “activityType”: “SEDENTARY”,
                “activeKilocalories”: 0,
                “steps”: 0,
                “distanceInMeters”: 0.0,
                “durationInSeconds”: 900,
                “activeTimeInSeconds”: 600,
                “met”: 1.0,
                “intensity”: “SEDENTARY”,
                “startTimeInSeconds”: 1454418900,
                “startTimeOffsetInSeconds”: 3600
        },
        {
                “summaryId”: “EXAMPLE_5678”,
                “activityType”: “RUNNING”,
                “activeKilocalories”: 257,
                “steps”: 427,
                “distanceInMeters”: 222.07,
                “durationInSeconds”: 900,
                “activeTimeInSeconds”: 300,
                “met”: 9.894117,
                “intensity”: “HIGHLY_ACTIVE”,
                “startTimeInSeconds”: 1454418900,
                “startTimeOffsetInSeconds”: 3600
        }
]
```

7.3. Sleep Summaries

Sleep summaries are data records representing how long the user slept and the automatically classified sleep levels during that sleep event (e.g. light, deep periods) based on data generated by the user's device.

Users may generate sleep data in three different ways. Some older Garmin devices (e.g. first generation vívofit) allow users to manually place the device in sleep mode. Newer devices do not have this option and instead, auto-detect sleep if it occurs between the user's Bed/Wake time range configured in Garmin Connect. Users may also self-report sleep information using Garmin Connect.

Sleep records from the Health API are labeled to identify how the sleep data was generated (see below). This allows partners to accept/reject various methods of collecting Sleep data. Recommended usage for this field is to filter out validation types that are not desired rather than accept only certain validation types in order to prevent lost data in the future if new validation types are added, as by default Garmin Connect displays records of all possible types.

Unlike Daily summaries which are associated with a given day on a midnight-to-midnight basis, Sleep summaries are associated with a user's overnight sleep range. Most will start on one calendar day and end on the next calendar day, but two different Sleep summaries can begin on the same day if, for example, the user goes to bed after midnight, wakes up, and then goes to bed before midnight the next evening.

Tip: Many Garmin devices attempt to auto-sync data during the night while the user is asleep, and the smartphone is charging. This may result in an incomplete Sleep summary record. It is important to update sleep data with the most recent data provided on subsequent ping notifications to get an accurate and full representation of the sleep window. See the "validation" parameter for more details.

Sleep levels from the Health API correspond to the sleep levels graph found in Garmin Connect. In both Garmin Connect and the Health API, the sleep summary will include REM sleep if the user's device is capable of REM sleep analysis. Users without REM-capable devices, or with REM-capable devices that have not been updated to REM-capable firmware, are limited to only deep, light, and awake sleep levels. Additionally, REM sleep will only be generated if the REM-capable device is set as the preferred activity tracker and is worn during sleep.

Some pulse-oximetry-enabled devices will generate SpO2 values during sleep for use in sleep analysis. If such values are generated, they are included in the sleep summary for reference.

Sleep score-enabled devices will generate sleep scores for use in sleep analysis if the user has the device set as the primary active tracker in the user's Garmin Connect account. If sleep scores are utilized by your application, please ensure any qualitative values are represented using the same descriptors provided through the API to avoid misleading or confusing End Users as described in the API License Agreement.

Each sleep summary may contain the following parameters:

Property	Type	Description
summaryId	string	Unique identifier for the summary.
calendarDate	string	The calendar date of this summary will be displayed on Garmin Connect. The date format is ‘yyyy-mm-dd’.
startTimeInSeconds	integer	Start time of the activity in seconds since January 1, 1970, 00:00:00 UTC (Unix timestamp).
startTimeOffsetInSeconds	integer	Offset in seconds to add to startTimeInSeconds to derive the “local” time of the device that captured the data.
durationInSeconds	integer	Length of the sleep monitoring period in seconds. (does not include awake or unmeasurable times)
totalNapDurationInSeconds	integer	Total time of nap duration for the monitoring period
unmeasurableSleepInSeconds	Integer	Time in seconds that the sleep level of the user could not be measured. This may or may not correspond to off-wrist time.
deepSleepDurationInSeconds	integer	Time in seconds the user spent in deep sleep during the sleep period.
lightSleepDurationInSeconds	integer	Time in seconds the user spent in light sleep during the sleep period.
remSleepInSeconds	integer	Time in seconds the user spent in REM sleep during the sleep period.
awakeDurationInSeconds	integer	Time in seconds the user spent awake during the sleep period.
sleepLevelsMap	Map	A map of sleep level time ranges, currently deep, light, and awake. Time ranges are represented as unix timestamps in seconds.
validation	string	The string that relays the validation state of the sleep data and its date range. The data could be auto-confirmed, but the sleep window could have been manually adjusted, or the sleep data itself is entirely manually entered. Possible values: MANUAL: The user entered sleep start and stop times manually through a web form. There is no device data backing up the sleep assessment. DEVICE: The user used a device with the sleep feature to manually start and stop sleep. This type still requires manual user intervention to judge sleep start and stop. OFF_WRIST: The device did not have enough heart rate data to make calculations for the sleep levels Map. (the device was off or too loose). Only start and end sleep times will be provided. AUTO_TENTATIVE: The sleep start and stop times were auto-detected by Garmin Connect using accelerometer data. However, further refinements to this sleep record may come later. This could be because the user is still asleep or because the user owns multiple devices and might sync another device later for this same time

		period. AUTO_FINAL: The sleep start and stop times were auto-detected by Garmin Connect, and enough data has been gathered to finalize the window. This status also indicates that the user only has one device so this record can never be updated again – users that own multiple devices will never get an AUTO_FINAL. AUTO_MANUAL: Sleep data was auto-detected by Garmin Connect, but the user is overriding the start and stop times or the user started with a manual entry and the sleep was auto-detected later. Garmin Connect stores both but will display the manual start and stop times in favor of the auto-detected times. ENHANCED_TENTATIVE: Sleep data was collected from a device capable of running an enhanced sleep analysis to detect REM sleep, but an updated sleep summary record may come later with further refinements or a greater sleep period. ENHANCED_FINAL: Sleep data was collected from a device capable of running an enhanced sleep analysis to detect REM sleep, and no further updates or refinements to this sleep analysis are expected.
timeOffsetSleepRespiration	Map	Collection of key-value pairs where the key is offset in seconds from the startTimeInSeconds and respiration measurement taken at that time. Respiration measurement is in breaths per minute.
timeOffsetSleepSpo2	Map	A map of SpO2 readings, where the keys are the offsets in seconds from the startTimeInSeconds and the values are the SpO2 measurements at that time. Only present if the user's device is SpO2-enabled.
overallSleepScore	Map	A map of overall sleep score, containing the quantitative value and the qualitative description of sleep.
sleepScores	Map	A map of sleep score string descriptions for each type of sleep as well as restless periods and stress levels during sleep. Each entry in the sleepScores will have a qualifier key value of EXCELLENT, GOOD, FAIR, or POOR that is used as a qualitative description of the user's period of sleep. Excellent: 90-100 Good: 80-89 Fair: 60-79 Poor: Below 60
naps	List	List of nap-related information recorded by device
napDurationInSeconds	Integer	Length of the monitoring period in seconds.
napStartTimeInSeconds	Integer	Start time of the activity in seconds since January 1, 1970, 00:00:00 UTC (Unix timestamp).
napValidation	String	The string that relays the validation state of the sleep data and its date range. Possible values:

		MANUAL: The user entered sleep start and stop times manually through a web form or mobile app. No device data is backing up the sleep assessment. DEVICE: The user used a device with nap tracking and the device detected the nap period.
napOffsetInSeconds	Integer	Offset in seconds to add to startTimeInSeconds to derive the “local” time of the device that captured the data.

Example

```
[
    {
        “summaryId”: “EXAMPLE_567890”,
        “calendarDate”: “2016-01-10”,
        “durationInSeconds”: 15264,
        “startTimeInSeconds”: 1452419581,
        “startTimeOffsetInSeconds”: 7200,
        “totalNapDurationInSeconds”: 600,
        “unmeasurableSleepDurationInSeconds”: 0,
        “deepSleepDurationInSeconds”: 11231,
        “lightSleepDurationInSeconds”: 3541,
        “remSleepInSeconds”: 0,
        “awakeDurationInSeconds”: 492,
        “sleepLevelsMap”: {
            “deep”: [
                {
                    “startTimeInSeconds”: 1452419581,
                    “endTimeInSeconds”: 1452478724
                }
            ],
            “light”: [
                {
                    “startTimeInSeconds”: 1452478725,
                    “endTimeInSeconds”: 1452479725
                }, {
                    “startTimeInSeconds”: 1452481725,
                    “endTimeInSeconds”: 1452484266
                }
            ]
        },
        “validation”: “DEVICE”
    },
    {
        “summaryId”: “EXAMPLE_567891”,
        “durationInSeconds”: 11900,
        “startTimeInSeconds”: 1452467493,
        “startTimeOffsetInSeconds”: 7200,
        “unmeasurableSleepDurationInSeconds”: 0,
        “deepSleepDurationInSeconds”: 9446,
        “lightSleepDurationInSeconds”: 0,
        “remSleepInSeconds”: 2142,
        “awakeDurationInSeconds”: 312,
        “sleepLevelsMap”: {
            “deep”: [
                {
                    “startTimeInSeconds”: 1452467493,
                    “endTimeInSeconds”: 1452476939
                },
             “light”: [
                {
                    “startTimeInSeconds”: 1452478725,
                    “endTimeInSeconds”: 1452479725
                }, {
                    “startTimeInSeconds”: 1452481725,
                    “endTimeInSeconds”: 1452484266
                }

            ],
            “rem”: [
                {
                    “startTimeInSeconds”: 1452476940,
                    “endTimeInSeconds”: 1452479082
                }
            ]
        },
        “validation”: “DEVICE”,
            “timeOffsetSleepRespiration”: {
            “60”: 15.31,
            “120”: 14.58,
            “180”: 12.73,
            “240”: 12.87
        },
        “timeOffsetSleepSpo2”: {
            “0”: 95,
            “60”: 96,
            “120”: 97,
            “180”: 93,
            “240”: 94,
            “300”: 95,
            “360”: 96
        },
        “overallSleepScore”: {
           “value”: 87,
           “qualifierKey”: “GOOD”
        },
        “sleepScores”: {
            “totalDuration”: {
                “qualifierKey”: “EXCELLENT”
            },
            “stress”: {
                “qualifierKey”: “EXCELLENT”
            },
                “awakeCount”: {
                     “qualifierKey”: “FAIR”
                },
                “remPercentage”: {
                     “qualifierKey”: “FAIR”
                },
                “restlessness”: {
                     “qualifierKey”: “GOOD”
                },
                “lightPercentage”: {
                     “qualifierKey”: “GOOD”
                },
                “deepPercentage”: {
                     “qualifierKey”: “POOR”
                }
           },
            "naps": [
                  {
                "napDurationInSeconds":600,
                "napStartTimeInSeconds": 1690916700,
                "napValidation": 'MANUAL'/'DEVICE',
                "napOffsetInSeconds": -18000
                  }


     ,
     {
           "summaryId": "x-EXAMPLE",
           "calendarDate": "2021-01-29",
           "durationInSeconds": 28260,
           "startTimeInSeconds": 1611840660,
           "startTimeOffsetInSeconds": 32400,
           "unmeasurableSleepInSeconds": 0,
           "deepSleepDurationInSeconds": 0,
           "lightSleepDurationInSeconds": 0,
           "remSleepInSeconds": 0,
           "awakeDurationInSeconds": 0,
           "validation": "OFF_WRIST",
           "timeOffsetSleepSpo2": {},
           "timeOffsetSleepRespiration": {}
     }

]
```

7.4. Body Composition Summaries

Body Composition summaries contain information about the user’s biometric data, like weight or body mass index. This data can be generated in two ways. Users can manually enter their weight on Garmin Connect. This results in a summary with only time and weight.

Finally, a user might have a Garmin Index body composition scale and sync data from this device. This will generate a summary with all possible biometric fields.

Each body composition summary may contain the following parameters:

Property	Type	Description
summaryId	string	Unique identifier for the summary.
measurementTimeInSeconds	integer	Time of measurement in seconds since January 1, 1970, 00:00:00 UTC (Unix timestamp).
measurementTimeOffsetInSeconds	integer	Offset in seconds to add to measurementTimeInSeconds to derive the “local” time of the device that captured the data.
muscleMassInGrams	integer	Muscle mass in grams.
boneMassInGrams	integer	Bone mass in grams.
bodyWaterInPercent	float	Percentage of body water (range 0.0 – 100.0).
bodyFatInPercent	float	Percentage of body fat. (range 0.0 – 100.0).
bodyMassIndex	float	Body mass index, or BMI.
weightInGrams	integer	Weight in grams.

Example

```
[
        {
                “summaryId”: “EXAMPLE_678901”,
                “measurementTimeInSeconds”: 1439741130,
                “measurementTimeOffsetInSeconds”: 0,
                “muscleMassInGrams”: 25478,
                “boneMassInGrams”: 2437,
                “bodyWaterInPercent”: 59.4,
                “bodyFatInPercent”: 17.1,
                “bodyMassIndex”: 23.2,
                “weightInGrams”: 75450
        },
        {
                “summaryId”: “EXAMPLE_678902”,
                “measurementTimeInSeconds”: 1439784330,
                “measurementTimeOffsetInSeconds”: 0,
                “muscleMassInGrams”: 25482,
                “boneMassInGrams”: 2434,
                “bodyWaterInPercent”: 59.8,
                “bodyFatInPercent”: 17.3,
                “bodyMassIndex”: 23.1,
                “weightInGrams”: 751732
        }
```

]

7.5. Stress Details Summaries

Stress Details summaries contain the user's stress level values for a given day. Stress levels are provided as 3-minute averages of the real-time stress scores generated on the device with values ranging from 1 to 100. A value of -1 means there was not enough data to detect stress, and -2 means there was too much motion (e.g. the user was walking or running).

Scores between 1 and 25 are considered "rest" (i.e. not stressful), 26-50 as "low" stress, 51-75 "medium" stress, and 76-100 as "high" stress. These numbers are derived based on a combination of many device sensors and will automatically adjust to the wearer of the device and gain accuracy over time as the stress algorithms learn the user's natural biometric norms. Stress values from the Health API are exactly the stress values shown on Garmin Connect.

Each stress details summary may contain the following parameters:

Property	Type	Description
summaryId	string	Unique identifier for the summary.
startTimeInSeconds	integer	Start time of the summary in seconds since January 1, 1970, 00:00:00 UTC (Unix timestamp).
startTimeOffsetInSeconds	integer	Offset in seconds to add to startTimeInSeconds to derive the "local" time of the device that captured the data.
durationInSeconds	integer	The duration of the measurement period in seconds.
calendarDate	string	The calendar date of this summary will be displayed on in Garmin Connect. The date format is 'yyyy-mm-dd'.
timeOffsetStressLevelValues	Map	Collection of mappings between offset from start time (in seconds) to a stress level value recorded for that time. Values correspond to Rest: < 26 Low: 26 – 50 Moderate: 51-75 High: 76 – 100 -1 -> OFF_WRIST -2 -> LARGE_MOTION -3 -> NOT_ENOUGH_DATA -4 -> RECOVERING_FROM_EXERCISE -5 -> UNIDENTIFIED
timeOffsetBodyBatteryValues	Map	Collection of mappings between offset from start time (in seconds) to a body battery value recorded for that time. Information on and a list of devices that support Body Battery are available here: https://support.garmin.com/ms-MY/?faq=2qczgfbN00AIMJbX33dRq9

bodyBatteryDynamicFeedbackEvent	Map	List of user’s current level body battery and time when it was calculated.
eventStartTimeInSeconds	Integer	Time of when body battery was calculated.
bodyBatteryLevel	String	Impact level from monitored events. Values correspond to Very Low: < 26 Low: 26 – 50 Moderate: 51-75 High: 76 - 100
bodyBatteryActivityEventList	List	List of events that affected the user’s body battery levels
eventType	String	Event type that contributed to user’s body levels changes. Possible fields are: “SLEEP”, “RECOVERY”, “NAP”, “ACTIVITY”, “STRESS”
eventStartTimeInSeconds	Integer	Start time of the summary in seconds since January 1, 1970, 00:00:00 UTC (Unix timestamp).
eventStartTimeOffsetInSeconds	Integer	Offset in seconds to add to startTimeInSeconds to derive the “local” time of the device that captured the data.
duration	Integer	Duration of the event
bodyBatteryImpact	Integer	Impact of the event on the user’s body battery. Positive numbers correspond to positive impact. Negative numbers correspond to negative impact.

Example

```
[
     {
          “summaryId”: “ EXAMPLE_6789124”,
          “calendarDate”: “2017-03-23”,
          “startTimeInSeconds”: 1490245200,
          “startTimeOffsetInSeconds”: 0,
          “durationInSeconds”: 540,
          “timeOffsetStressLevelValues”: {
               “0”: 18,
               “180”: 51,
               “360”: 28,
               “540”: 29
          },
          “timeOffsetBodyBatteryValues”: {
                     “0”: 55,
                     “180”: 56,
                     “360”: 59
          },
          "bodyBatteryDynamicFeedbackEvent": {
                    "eventStartTimeInSeconds": 1692693913,
                    "bodyBatteryLevel": "VERY_LOW"
          },
           "bodyBatteryActivityEvents": [
                       {
                           "eventType": "SLEEP",
                           "eventStartTimeInSeconds": 1692673020,
                           "eventStartTimeOffsetINSeconds": -18000,
                           "duration": 30840,
                           "bodyBatteryImpact": 25
                       },
                       {
                           "eventType": "RECOVERY",
                           "eventStartTimeInSeconds": 1692725550,
                           "eventStartTimeOffsetInSeconds": -18000,
                           "duration": 1680,
                           "bodyBatteryImpact": 3
                       }

]
```

7.6. User Metrics Summaries

User Metrics are per-user calculations performed by Garmin based on the underlying data uploaded from the user's device. This data can be specific to a single device and field availability is dependent on device model support. More information about Fitness age can be found at https://support.garmin.com/en-US/?faq=CM1YJmMrrNAbEpM9PapJ07.

Unlike other summaries, User Metrics are associated only with a calendar date, not a specific time frame, and only the most recent value for any fields is presented to the user.   Each metric directly corresponds to the similarly named field found in Garmin Connect.

Each user metrics summary may contain the following parameters:

Property	Type	Description
summaryId	string	Unique identifier for the summary.
calendarDate	string	The calendar date of this summary will be displayed on Garmin Connect. The date format is 'yyyy-mm-dd'.
vo2Max	float	An estimate of the maximum volume of oxygen (in milliliters) the user can consume per minute per kilogram of body weight at maximum performance.
vo2MaxCycling	float	An estimate of the maximum volume of oxygen for Cycling activities (in milliliters) the user can consume per minute per kilogram of body weight at maximum performance. This field will be included only if data is available.
enhanced	boolean	When set to true, the Fitness Age provided has been calculated using a new algorithm (taking into account activity intensity, resting heart rate, and body fat percentage or BMI). When set to false, the value provided for Fitness Age has been calculated using the older method of estimation. More information on the improved Fitness Age calculation and device compatibility can be found here.
fitnessAge	integer	An estimation of the 'age' of the user's fitness level, is calculated by comparing internal fitness metrics with the average readings of biometrically similar users by age. For instance, a fitness age of 48 indicates that the user's physical fitness is similar to that of an average 48-year-old person of the same gender. Improved Fitness Age (enhanced =true) takes into account activity intensity, resting heart rate, and body fat percentage or BMI.

Example

```
[
  {
    "summaryId": " EXAMPLE_843244",
    "calendarDate": "2017-03-23",
    "vo2Max": 48.0,
    "enhanced": true
    "fitnessAge": 32
    }
]
```

7.7. Pulse Ox Summaries

Pulse Ox summaries contain blood oxygen saturation data. Two types of data are represented in Pulse Ox summaries based on the capabilities of the user's device. If the onDemand field is set to false, the timeOffsetSpo2Values map contains a SpO2 measurement that is an average of all measurements taken as part of the Acclimation feature (https://www8.garmin.com/manuals/webhelp/fenix5plus/EN-US/GUID-4D425925-D4EE-4C26-B974-5375D0670860.html). If the onDemand field is true the timeOffsetSpo2Values map instead contains one or more exact measurements taken by a device that is capable of on-demand measurements but not the Acclimation feature, such as the Vivosmart 4. The *durationInSeconds* field will always be 0, for onDemand measurements summaries.

Tip: If user was tracking Pulse Ox using Pulse Ox all day feature and Pulse Ox on demand (Spot check), 2 separate summaries will be generated reflecting each measurement type.

Backfill is supported for both Pulse Ox summary types (all-day measurements and On-Demand measurements)

Each Pulse Ox summary may contain the following parameters:

Property	Type	Description
summaryId	string	Unique identifier for the summary.
calendarDate	string	The calendar date of this summary will be displayed on in Garmin Connect. The date format is 'yyyy-mm-dd'.
startTimeInSeconds	float	Start time of the summary in seconds since January 1, 1970, 00:00:00 UTC (Unix timestamp).
startTimeOffsetInSeconds	integer	Offset in seconds to add to startTimeInSeconds to derive the "local" time of the device that captured the data.
durationInSeconds	integer	The duration of the measurement period in seconds.
timeOffsetSpo2Values	Map	Collection of key-value pairs where the key is offset in seconds from the startTimeInSeconds and the value is the SpO2 measurement taken at that time (1 sample/minute)
onDemand	boolean	A Boolean to show whether this pulse ox summary represents an on-demand reading or an averaged acclimation reading.

Example

```
[
      {
            “summaryId”:”Example1234”,
            “calendarDate”:”2018-08-27”,
            “startTimeInSeconds”:1535400706,
            “durationInSeconds”:86400,
            “startTimeOffsetInSeconds”:3600,
            “timeOffsetSpo2Values”:  {
                  “7140”:94,
                  “10740”:98,
                  “10800”:99,
                  “10860”:98,
                  “10920”:98,
                  “10980”:97,
                  “11040”:97,
                  “11100”:98,
                  “11160”:97,
                  “11220”:96,
                  “11280”:96,
                  “11340”:97,
                  “11400”:97,
                  “11460”:96,
                  “11520”:96,
                  …
                  “75540”:95,
                  “79140”:96,
                  “82740”:97,
                  “86340”:96
            },
            “onDemand”:false
      },
      {
            “summaryId”:”example1234-spo2OnDemand”,
            “calendarDate”:”2018-08-27”,
            “startTimeInSeconds”:1572303600,
            “durationInSeconds”:0,
            “startTimeOffsetInSeconds”:3600,
            “timeOffsetSpo2Values”:  {
                  “55740”:93
      },
      “onDemand”:true
      }
      ]
```

7.8. Respiration Summaries

Respiration is a feature (https://www8.garmin.com/manuals/webhelp/vivoactive4_4S/EN-US/GUID-252F74B6-C24B-495B-8E73-4BD595CA7FE3.html) available on some Garmin devices that tracks breathing rate throughout the day, during sleep, and during activities such as breathwork and yoga.

Each Respiration summary may contain the following parameters:

Property	Type	Description
summaryId	string	Unique identifier for the summary.
startTimeInSeconds	float	Start time of the summary in seconds since January 1, 1970, 00:00:00 UTC (Unix timestamp).
durationInSeconds	integer	The duration of the measurement period in seconds.
startTimeOffsetInSeconds	integer	Offset in seconds to add to startTimeInSeconds to derive the “local” time of the device that captured the data.
timeOffsetEpochToBreaths	Map	Collection of key-value pairs where the key is offset in seconds from the startTimeInSeconds and respiration measurement taken at that time. Respiration measurement is in breaths per minute.

Example

```
[
    {
        “summaryId”: “x15372ea-5d7866b4”,
        “startTimeInSeconds”: 1568171700,
        “durationInSeconds”: 900,
        “startTimeOffsetInSeconds”: -18000,
        “timeOffsetEpochToBreaths”: {
            “0”: 14.63,
            “60”: 14.4,
            “120”: 14.38,
            “180”: 14.38,
            “300”: 17.1,
            “540”: 16.61,
            “600”: 16.14,
            “660”: 14.59,
            “720”: 14.65,
            “780”: 15.09,
            “840”: 14.88
        }
    },
    {
        “summaryId”: “x15372ea-5d786a38”,
        “startTimeInSeconds”: 1568172600,
        “durationInSeconds”: 900,
        “startTimeOffsetInSeconds”: -18000,
        “timeOffsetEpochToBreaths”: {
            “0”: 14.82,
            “60”: 16.58,
            “120”: 13.2,
            “180”: 14.69,
            “240”: 16.17,
            “300”: 16.04,
            “540”: 13.82,
            “600”: 13.26,
            “660”: 12.76,
            “780”: 13.3,
            “840”: 13.53
        }
    }
]
```

7.9. Health Snapshot Summaries

The Garmin Health Snapshot is a collection of key health-related insights recorded during a two-minute session on a compatible device. Heart rate (HR), heart rate variability (HRV), Pulse Ox, respiration, and stress are the metrics included this summary, which collectively provide you a glimpse of your overall cardiovascular status. More information about Health Snapshot can be found at https://support.garmin.com/en-US/?faq=PB1duL5p6V64IQwhNvcRK9.

Each Health Snapshot summary may contain the following parameters:

Property	Type	Description
summaryId	string	Unique identifier for the summary.
calendarDate	string	The calendar date of this summary will be displayed on Garmin Connect. The date format is 'yyyy-mm-dd'.
startTimeInSeconds	float	Start time of the summary in seconds since January 1, 1970, 00:00:00 UTC (Unix timestamp).
durationInSeconds	integer	The duration of the measurement period in seconds.
startTimeOffsetInSeconds	integer	Offset in seconds to add to startTimeInSeconds to derive the "local" time of the device that captured the data.
summaries	List	List of summary types and their corresponding data related to Health Snapshot. Summary types included in this list include heart rate, stress, pulse ox, respiration, SDRR, and RMSSD.

```
[{
     "summaryId": "x42f72c9-612e11dae53d462a-0b98-4ae8-9fdc-
28f392a1cd8078",
     "calendarDate": "2021-08-31",
     "startTimeInSeconds": 1630409178,
     "durationInSeconds": 120,
     "offsetStartTimeInSeconds": 7200,
     "summaries": [{
                "summaryType": "heart_rate",
                "minValue": 78.0,
                "maxValue": 87.0,
                "avgValue": 83.0,
                "epochSummaries": {
                     "0": 84.0,
                "1": 84.0,
                "2": 83.0,
                "3": 83.0,
                "4": 83.0,
                "5": 84.0,
                "115": 82.0,
                "116": 82.0,
                "117": 83.0,
                "118": 85.0,
                "119": 85.0,
                "120": 85.0
            }
        },
        {
            "summaryType": "respiration",
            "minValue": 13.449999809265137,
            "maxValue": 15.319999694824219,
            "avgValue": 14.489999771118164,
            "epochSummaries": {
                "0": 15.319999694824219,
                "1": 15.319999694824219,
                "2": 15.319999694824219,
                "3": 15.319999694824219,
                "4": 15.09000015258789,
                "5": 15.09000015258789,
                "115": 13.859999656677246,
                "116": 13.859999656677246,
                "117": 14.300000190734863,
                "118": 15.229999542236328,
                "119": 15.229999542236328,
                "120": 15.319999694824219
            }
        },
        {
            "summaryType": "stress",
            "minValue": 78.0,
            "maxValue": 87.0,
            "avgValue": 82.0,
            "epochSummaries": {
                "0": 78.0,
                "1": 78.0,
                "2": 78.0,
                "3": 78.0,
                "4": 78.0,
                "5": 78.0,
                "115": 83.0,
                "116": 83.0,
                "117": 83.0,
                "118": 82.0,
                "119": 82.0,
                "120": 82.0
            }
        },
        {
            "summaryType": "spo2",
            "minValue": 84.0,
            "maxValue": 86.0,
            "avgValue": 85.0,
            "epochSummaries": {
                "0": 86.0,
                "1": 86.0,
                "2": 86.0,
                "3": 86.0,
                "4": 86.0,
                "5": 86.0,
                "115": 84.0,
                "116": 84.0,
                "117": 84.0,
                "118": 86.0,
                "119": 86.0,
                "120": 86.0
            }
        },
        {
            "summaryType": "rmssd_hrv",
            "avgValue": 20.0
        },
        {
            "summaryType": "sdrr_hrv",
            "avgValue": 32.0
        }
    ]
}]
```

7.10. Heart Rate Variability (HRV) Summaries

Heart rate variability (HRV) refers to beat-to-beat variations in heart rate and is data collected during the overnight sleep window for select devices. To gain a deeper understanding of your overall health, recovery, and training performance through heart rate variability while you sleep, based on technology developed by our Firstbeat Analytics™ team, please visit https://discover.garmin.com/en-US/performance-data/running/#heart-rate-variability.

Each HRV summary may contain the following parameters:

Property	Type	Description
summaryId	string	Unique identifier for the summary.
calendarDate	string	The calendar date this summary

		would be displayed on in Garmin Connect. The date format is ‘yyyy-mm-dd’.
startTimeInSeconds	float	Start time of the summary in seconds since January 1, 1970, 00:00:00 UTC (Unix timestamp).
durationInSeconds	integer	The duration of the measurement period in seconds.
startTimeOffsetInSeconds	integer	Offset in seconds to add to startTimeInSeconds to derive the “local” time of the device that captured the data.
lastNightAvg	integer	The average heart rate variability value from the last night of data.
lastNight5MinHigh	integer	The maximum HRV value over any 5 minute interval of the last night of data.
hrvValues	Map	A map of the HRV values and the time offset of when each value was recorded. Lack of entry for a given offset should be interpreted as no data available. rmssd

```
[{
     "summaryId": "x473db21-6295abc4",
     "calendarDate": "2022-05-31",
     "lastNightAvg": 44,
     "lastNight5MinHigh": 72,
     "startTimeOffsetInSeconds": -18000,
     "durationInSeconds": 3820,
     "startTimeInSeconds": 1653976004,
     "hrvValues": {
          "300": 32,
          "600": 24,
          "900": 31,
          "1200": 35,
          "1500": 39,
          "1800": 47,
          "2100": 32,
          "2400": 24,
          "2700": 31,
          "3000": 35,
          "3300": 39,
          "3600": 47
     }
}]
```

7.11. Blood Pressure Summaries

Blood pressure summaries offer data from blood pressure readings taken using an Index™ BPM device or from a user's manually uploaded blood pressure data. This includes systolic, diastolic, and pulse values taken at the time of the blood pressure reading. For more information about the validation of data using the Index™ BPM, please visit https://www.garmin.com/en-US/bpmvalidation/.

Each blood pressure summary may contain the following parameters:

Property	Type	Description
summaryId	string	Unique identifier for the summary.
measurementTimeInSeconds	integer	Measurement time of the summary in seconds since January 1, 1970, 00:00:00 UTC (Unix timestamp).
measurementTimeOffsetInSeconds	integer	Offset in seconds to add to measurementTimeInSeconds to derive the "local" time of the device that captured the data.
systolic	integer	The systolic value of the blood pressure reading.
diastolic	integer	The diastolic value of the blood pressure reading.
pulse	integer	Pulse rate at the time the blood pressure reading.
sourceType	string	This field is used to determine if blood pressure data was entered manually or synced from a Garmin Device. Possible values: MANUAL: The user entered blood pressure information manually through a web form. DEVICE: The user used a Garmin device to perform a blood pressure reading.

```
[{
        "summaryId": "x473db21-632b3500",
        "systolic": 120,
        "diastolic": 110,
        "pulse": 82,
        "sourceType": "MANUAL",
        "measurementTimeInSeconds": 1663776000,
        "measurementTimeOffsetInSeconds": -18000
} ]
```

7.12. Skin Temperature

While wearing your compatible Garmin® watch to sleep, you can see skin temperature changes during your sleep window. Changes in skin temperature can be related to activity, illness, and other factors.

Note: Not all Garmin devices support recording of sleep skin temperature.
Garmin Connect Feature overview HERE.

The JSON model contains the following data fields:

Property	Type	Description
summaryId	string	Identifier of the summary.
calendarDate	string	The calendar date of this summary will be displayed on Garmin Connect. The date format is ‘yyyy-mm-dd’.
avgDeviationCelsius	float	Average deviation of user’s body temperature for the monitoring period.
durationInSeconds	integer	Length of the monitored period in UTC timestamp in seconds.
startTimeInSeconds	integer	Start time of the activity in seconds since January 1, 1970, 00:00:00 UTC (Unix timestamp).
startTimeOffsetInSeconds	integer	Offset in seconds to add to startTimeInSeconds to derive the “local” time of the device that captured the data.

Example:

```
{
"summaryId": “example-65f83c38",
 "calendarDate": "2024-03-18",
 "avgDeviationCelsius": -1.6,
 "durationInSeconds": 1980,
 "startTimeInSeconds": 1710767160,
 "startTimeOffsetInSeconds": -21600
}
```

8. Summary Backfill

This service provides the ability to request historic summary data for a user. Historic data, in this context, means any data uploaded to Garmin Connect before the user's registration with the partner program, or any data that has been purged from the Health API due to the data retention policy.

A backfill request returns an empty response immediately, while the actual backfill process takes place asynchronously in the background. Once backfill is complete, a notification will be generated and sent as if data for that period was newly synced. Both the Ping Service and the Push Service are supported by Summary Backfill. The maximum date range (inclusive) for a single backfill request is 90 days, but it is permissible to send multiple requests representing other 90-day periods to retrieve additional data.

Evaluation keys are rate-limited to 100 **days** of data backfilled per minute rather than by total HTTP calls performed. For example, two backfill requests for 60 days of data would trigger the rate limit, but twenty calls for three days of data would not.

Production keys have the following rate limit: 10,000 days/data requested per minute.

Per user rate limit: 1 month since the first user connection per summary type.
Note: Duplicate Backfill requests are rejected with HTTP 409 status (duplicate requests – requests for already requested time)

Request

Resource URL for daily summaries
GET https://apis.garmin.com/wellness-api/rest/backfill/dailies

Resource URL for epoch summaries
GET https://apis.garmin.com/wellness-api/rest/backfill/epochs

Resource URL for sleep summaries
GET https://apis.garmin.com/wellness-api/rest/backfill/sleeps

Resource URL for body composition summaries
GET https://apis.garmin.com/wellness-api/rest/backfill/bodyComps

Resource URL for stress details summaries
GET https://apis.garmin.com/wellness-api/rest/backfill/stressDetails

Resource URL for user metrics summaries
GET https://apis.garmin.com/wellness-api/rest/backfill/userMetrics

Resource URL for Pulse Ox summaries
GET https://apis.garmin.com/wellness-api/rest/backfill/pulseOx

Resource URL for Respiration summaries
GET https://apis.garmin.com/wellness-api/rest/backfill/respiration

Resource URL for Health Snapshot Summaries
GET https://apis.garmin.com/wellness-api/rest/backfill/healthSnapshot

Resource URL for HRV Summaries
*GET **https://apis.garmin.com/wellness-api/rest/backfill/hrv***

Resource URL for Blood Pressure Summaries
GET https://apis.garmin.com/wellness-api/rest/backfill/bloodPressures

Resource URL for Skin Temperature summaries
*GET **https://apis.garmin.com/wellness-api/rest/backfill/skinTemp***

Request parameters

Parameter	Description
summaryStartTimeInSeconds	A UTC timestamp represents the beginning of the time range to search based on the moment the data was recorded by the device. This is a required parameter.
summaryEndTimeInSeconds	A UTC timestamp represents the end of the time range to search based on the moment the data was recorded by the device. This is a required parameter.

Response

Since backfill works asynchronously, a successful request returns HTTP status code 202 (accepted) with no response body. Please see Appendix E for possible error responses.

Example

Request:
GET https://apis.garmin.com/wellness-api/rest/backfill/dailies?summaryStartTimeInSeconds=1452384000&summaryEndTimeInSeconds=1453248000

This request triggers the backfill of daily summary records which were recorded in the time between UTC timestamps 1452384000 (2016-01-10, 00:00:00 UTC) and 1453248000 (2016-01-20, 00:00:00 UTC).

9. Requesting a Production Key

The first consumer key generated through the Developer Portal is an evaluation key.
This key is rate-limited and should only be used for testing, evaluation, and development. *Evaluation-level apps that violate API guidelines may be disabled without prior notice***.** To obtain a production-level key, your integration must pass the technical and UX review. Garmin must approve and review the API integration to ensure a high-quality user experience and compliance with the brand guidelines.

Production Review:

To initiate the review, please contact connect-support@developer.garmin.com.

1. Technical Review:

Please provide a screenshot or complete verification. You can use the Partner Verification tool to ensure that the following technical requirements are met:

- Authorization for at least two Garmin Connect users
- User Deregistration/User Permission endpoints enabled
- PING/PUSH notification processing (PULL-ONLY requests not allowed)
- HTTP 200 sent asynchronously within 30 seconds to all data received (min payload allowed 10MB, Activity Details: 100MB)

2. UX and Brand Compliance Review:

To ensure the user experience and branding comply with Garmin's guidelines, **submit screenshots and/or video demonstrating**:

- All uses of Garmin trademarks, logos, and brand elements throughout the app
- All instances of Garmin products and imagery
- All required attribution statements, as specified in the API brand guidelines
- A complete view of the user experience (UX) flow, ensuring Garmin is accurately represented and not mischaracterized

Note: All instances where Garmin branding, marks, or attribution appear in the app must be included in the submission

3. Account Set up

- All authorized users were added to the account (see Section 4 of the Start Guide).
- Signed up for the API Blog email to be aware of future changes.

Appendix A – Activity Types

Below is the list of valid activity types referenced in EPOCH summaries.

Activity	Description
WALKING	Steps recorded
RUNNING	Steps recorded above walking tolerance
WHEELCHAIR_PUSH	Pushes recorded
SEDENTARY	Little to no activity monitored (low movement, sitting, resting, or sleeping)
GENERIC	No steps recorded, higher heart rate
SLEEP	Recorded ONLY by Vivofit device when in sleep mode.

Appendix B – Wellness Monitoring Intensity

Below is the list of possible intensity values for wellness monitoring summaries.

Monitoring Intensity	Notes
SEDENTARY	Little to no activity monitored. This could be due to minimal movement, sitting, resting, or sleeping.
ACTIVE	Some activity was monitored. A brisk walk could achieve this intensity.
HIGHLY_ACTIVE	High activity monitored. Running or speed walking could achieve this intensity.

Appendix C – MET Value

Metabolic Equivalent of Task (MET) is an official measure of activity intensity. Garmin's calculation of MET is an estimation based on the biometric data provided (height, weight, date of birth, gender) and improves in accuracy if heart rate data is also captured. The following linked document hosted by the US Centers for Disease Control and Prevention provides detailed information on MET and physical activity intensity: http://www.cdc.gov/nccdphp/dnpa/physical/pdf/PA_Intensity_table_2_1.pdf

Appendix D – Motion Intensity

Motion Intensity is a numerical abstraction of low-level accelerometer data, provided for use in further analysis. This data is not exposed directly to the consumer by Garmin but is used in the creation of other metrics. Motion Intensity is calculated at minute-level granularity as a number between 0 and 7, with 0 being still and 7 being constant, sharp motion. Unlike steps, distance, or activity type, which take net movement into account, motion intensity will increase even if the user does not move in space. For instance, if a user were to jump up and down or fidget with a pencil they would not get credit for any distance, but their motion intensity scores for that monitoring period would increase. It is very common to see mid-range max motion intensities even for sedentary epochs as most people do not sit still.

Appendix E – Error Responses

Usually, the service responds to all requests with HTTP status code 200 (OK). In case of an error, one of the following HTTP status codes may be sent. When any of these HTTP status codes are present, the response body will contain a JSON object with an error message to assist in isolating the exact reason for the error in the following form:

{ "errorMessage": "The error message details" }

HTTP status code	Description
400 - Bad Request	One of the input parameters is invalid. See the error message in the response body for details.
401 - Unauthorized	The authorization for the request failed. See the error message in the response body for details.
403 - Forbidden	The User Access Token in the request header is unknown. This could be the result of a malformed token or a token that has been invalidated by the user removing their consent from the Garmin Connect account page.
412 - Precondition failed	The User Access Token is valid, but the user has not given his permission for the summary type on the Garmin Connect account page. Other summary types might still work since the user didn't remove his consent in general (API toggle is turned off)
500 - Internal Server Error	Any server error that does not fall into one of the above categories.

Example

Request:

GET https://apis.garmin.com/wellness-api/rest/epochs?uploadStartTimeInSeconds=1452384000&uploadEndTimeInSeconds=1452777797000

Response:

```
HTTP/1.1 400 Bad Request
Date           Wed, 03 Feb 2016 12:15:17 GMT
Server         Apache
Content-Length 118
Content-Type   application/json;charset=utf-8

{
      "errorMessage": "timestamp '1452777797000' appears to be in
milliseconds. Please provide unix timestamps in seconds."
}
```