GARMIN INTERNATIONAL

Health REST API Specification

Version 2.9.5

CONFIDENTIAL

Table of Contents

1 Revision History

Version	Date	Revisions
2.6	6/12/2018	• REM sleep • userId in Ping and Push notifications
2.6.1	6/27/2018	Specifying new Enhanced Sleep validation types for clarity
2.6.2	7/27/2018	Various grammar and branding corrections.
2.7	8/28/2018	New summary type: Pulse Ox
2.8	10/9/2018	New fields: Body Battery in Stress and Pulse Ox in Sleep
2.8.1	11/15/2018	New field: Laps in Activity Details.
2.9	12/14/2018	Modifications to Pulse Ox data model and support for on-demand SpO2 measurements.
2.9.2	1/17/2019	New fields in Activity Details
2.9.3	4/10/2019	New data type: Activity Files
2.9.4	4/15/2019	Backfill for Activity Details and Files enabled. All ‘activity’ types are backfilled using the same activities backfill endpoint documented below.
2.9.5	9/24/2019	New data type: Menstrual Cycle Tracking (MCT)

2 Getting Started

2.1 Purpose of the API

The Garmin Health API is the mechanism by which Garmin users can share the data they generate on their activity trackers and fitness devices with non-Garmin corporate partners. The Health API is not meant to be a long-term data store that corporate partners can call to fetch historical data on demand. Instead, a Garmin Health API implementation should fetch data from the Health API as soon as a notification of new data is received, and if desired, persist it locally within the partner infrastructure for long-term storage.

Data will be removed from the Health API fifteen days after it is initially loaded by a user device sync or when the user removes their consent to share their data with the partner, whichever comes first. The removed data is then considered historical and may be reloaded through the Backfill process (see Summary Backfill below) .

2.2 Consumer Key and Secret

Garmin Health partners will be provided with a consumer key and secret used to gain access to the Health REST API. The consumer key is used to uniquely identify a partner and the consumer secret is used to validate that the requests received are from that partner and not a third-party that has gained unauthorized access to the consumer key.

The consumer key can be considered public information, but the consumer secret is private. For the security of users, the consumer secret should be secured and never sent over a network in plain text. It is not permitted to embed the consumer secret into consumer products like mobile apps.

Consumer key credentials are created using the Developer Portal and creating *Apps* (https://developerportal.garmin.com/developer-programs/content/56/overview). Each app represents a unique Consumer Key. Your first app will generate an evaluation-level consumer key that is rate-limited. Once your integration has been verified for product, subsequent apps will create consumer keys with production-level access. Please see "Requesting a Production Key" below for more information.

Note:
Multiple consumer keys should be created to correspond to projects or implementations whose user base is logically separated. A common scenario is for one partner to manage user data from multiple other companies. A new key should be created and associated with each managed company so that Garmin users can make an informed decision to consent to sharing their data with just that company.

2.3 User Registration

Before a partner can access a user's data, the user must grant the partner access. Please refer to the detailed Garmin OAuth documentation for details on acquiring, authorizing, and signing with a User Access Token (UAT) to access Garmin

user data. The Health API web tools (see Web Tools below) also contain additional demonstrations of user authorization and request signing.

2.4 Health API Integration Strategies

A real-time integration must be demonstrated in order to gain production-level access to the Health API. Partners have two options for real-time integrations: Push integration or Ping/Pull integration. In both cases HTTP POSTs are sent by Garmin systems in JSON format for each user and each summary type. Ping notifications contain callback URLs that, when invoked, will return the user's summary data, while Push notifications contain the same summary data within the POST body itself and require no callback. All data must be consumed through one of these two mechanisms, but partners may choose freely which one suits their business and technical needs better. See detailed information on the Ping Service and Push Service below.

2.5 Summary Data

All data available from the Health API is categorized as different types of summary data. Push integrations receive this data directly from the Push notification POST body, but Ping/Pull integrations must call the Health API with requests signed with the Consumer Key (representing the partner) and the UAT (representing the user) via OAuth. The summary data should be archived by the partner, as the Health API only keeps user data for fifteen days from the upload date.

Tip: The Health API does not automatically provide data uploaded to Garmin Connect before the UAT was issued, but historic summary information is available using the Backfill Service or Web Tool (see Backfill below).

2.6 Summary Data Types

Health API integrations may choose to consume any combination of different summary types, with summary data being categorized based on the data and timespan it represents.

- **Dailies**
  Daily summaries contain wellness data on a daily basis such as steps, distance, and heart rate. This summary data type should be used to get daily average representations of a user's day, similar to the "My Day" view in Garmin Connect.
- **Activities (Summary Only)**
  Activity summaries provide high-level information about discrete fitness activities such as running, swimming, or biking. These summaries are generated when a user records a timed activity on their device. Use this summary type only if additional information about an activity (e.g. average heart rate or calories burned) is required, as wellness data (such as steps and distance) is already included in the Dailies and Epochs summaries.
- **Manually Updated Activities**
  Manually Updated Activities are fitness activities that have been changed or created by the user and may not represent data recorded by a device. A user has either manually edited the details or created the activity from scratch using Garmin Connect. Garmin Connect allows all activity fields to be edited. This summary type is provided to allow partners to purposefully include or exclude user-edited activities in their program.

- **Activity Details**
  Activity Details contain detailed Fitness Activity data generated by the user's device during discrete fitness activities, including GPS coordinates and all available sensor data. These summaries are generated when a user

records a timed activity on their device. Activity Details summaries are considered a premium summary type and are not available by default of newly-created applications. To gain access to Activity Details data, please contact the Health API Support team (support@health.garmin.com).

- **Epochs**
  Epoch summaries contain information about wellness data, such as steps and distance, broken down in to short time periods (15 minutes). This summary type should be used to construct a more detailed representation of the user's activity within a day, such as the "Step Details" chart in Garmin Connect.
- **Sleep**
  Sleep summaries provide data about sleep duration and sleep level classification, including light, deep, REM, and awake periods. This data can be set manually by the user or auto-detected by the device, but only during the sleep range the user has pre-configured in Garmin Connect.
- **Body Composition**
  Body Composition summaries provide metrics like weight, body mass index, and muscle mass.
- **Detailed Stress**
  Detailed Stress summaries provide averaged stress level scores with 3-minute granularity from supported devices. Stress level scores range from 1 to 100.
- **User Metrics**
  User Metric summaries are data calculated algorithmically on a per-user basis, such as fitness age or estimated VO2 max.
- **Move IQ Events**
  Move IQ Event summaries are a feed of activities which have been automatically detected by the device based on movement patterns, like running or biking. Wellness data from Move IQ events is already included in the Daily and Epoch summaries. For more details about Move IQ events, see https://support.garmin.com/faqSearch/en-US/faq/content/zgFpy8MShkArqAxCug5wC6.
- **Pulse Ox**
  Pulse Ox summaries contain pulse oxygen (SpO2) data. Both on-demand pulse ox data and pulse ox measurements powering the Acclimation feature are included.
- **Deregistration**
  Not a traditional summary type, these are one-way records indicating that a user has removed their consent to share their data.

A full list of Summary data endpoints and descriptions of each Summary data type are provided below.

2.7 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. To obtain a production-level key that is not rate-limited, your integration must be verified using the Partner Verification Tool.

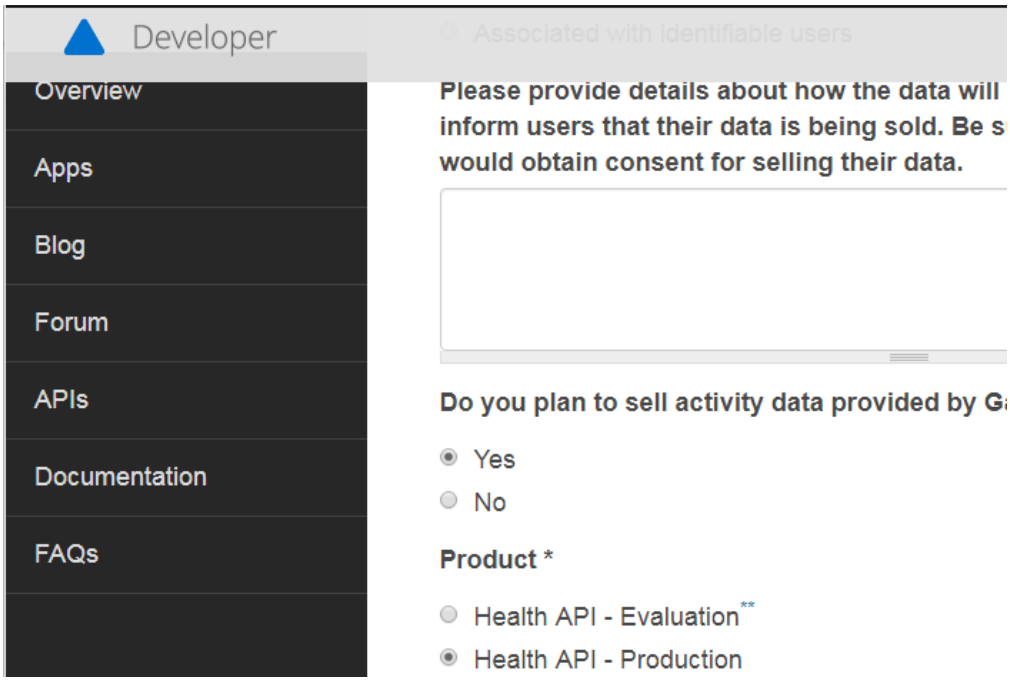
Tip: Before requesting a production key, please make sure your integration meets these basic requirements:

- 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.

1. Access the Partner Verification Tool (https://healthapi.garmin.com/tools/partnerVerification) and use your existing evaluation key.

2. Click *Run Tests* to start the automatic verification. The tool will perform a series of integration tests and checks. If all requirements have been met, you may request a production key using the Developer Portal.

3. In the Developer Portal, click on "Apps" and ("+Add a New App) to load the Add App form. When completing the form, choose "Health API – Production" under Product (see image below). A member of the Health API support team will approve the Production key request as soon as possible.



3 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 that haven't synced any new data.

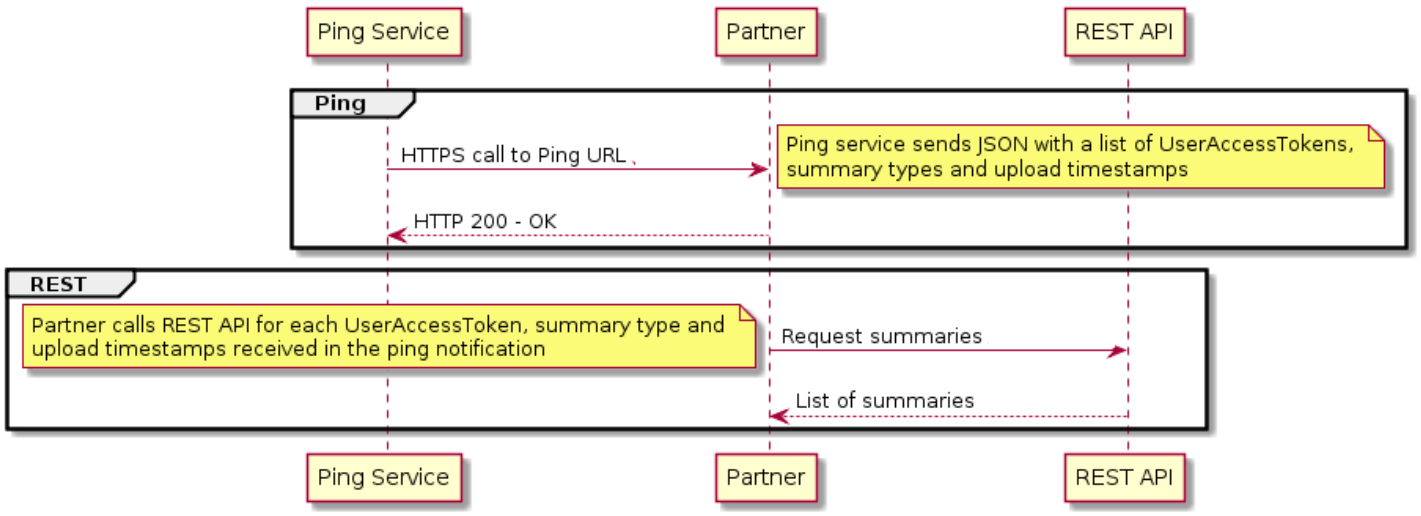
Each notification also contains a callback URL. When this URL is called, data specific to that user and summary type is returned. The partner may provide separate URLs for each summary type for flexible processing or may choose to send ping notifications for all data types to the same endpoint.

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 UATs 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.

3.1 Ping Workflow

The following diagram illustrates the general workflow.



The Ping Service has a timeout of thirty seconds. In order 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 on a regular basis. 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 (support@health.garmin.com). They will be happy to help set up a regeneration of all missed notifications during the affected time.

3.2 Ping Notification Content

JSON Element	Description
summary type (list key)	The summary type of this list of pings
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. However, it will persist across userAccessTokens should the user re-register to generate a new UAT.
userAccessToken	The UAT for which new data is available
uploadStartTimeInSeconds	The upload start timestamp of the new data in seconds since January 1, 1970, 00:00:00 UTC (Unix timestamp). Not present for deregistration notifications.
uploadEndTimeInSeconds	The upload end timestamp of the new data in seconds since January 1, 1970, 00:00:00 UTC (Unix timestamp). Not present for deregistration notifications.
callbackURL	Pre-formed URL to pull the data. Not present for deregistration notifications.

Example


```
{
      "epochs": [{
            "userId": "4aacafe82427c251df9c9592d0c06768",
            "userAccessToken": "8f57a6f1-26ba-4b05-a7cd-c6b525a4c7a2",
            "uploadStartTimeInSeconds": 1444937651,
            "uploadEndTimeInSeconds": 1444937902,
            "callbackURL": "https://healthapi.garmin.com/wellness-
api/rest/epochs?uploadStartTimeInSeconds=1444937651&uploadEndTimeInSeconds=144493
7902"
      }]
}
```


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.

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

```
curl -v -X POST -H "Content-Type: application/json;charset=utf-8" -d
'{"epochs":[{"userAccessToken":"8f57a6f1-26ba-4b05-a7cd-
c6b525a4c7a2","uploadStartTimeInSeconds":1444937651,"uploadEndTimeInSeconds":1444
937902,"callbackURL":"https://healthapi.garmin.com/wellness-
api/rest/epochs?uploadStartTimeInSeconds=1444937651&uploadEndTimeInSeconds=144493
7902"}]}' http://localhost:8080/garmin/ping
```

4 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 would have been returned by the Health API 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.

4.1 Push Notification Content

JSON Element	Description
summary type (list key)	The summary type of this list of pings.
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. However, it will persist across userAccessTokens should the user re-register to generate a new UAT.
userAccessToken	The UAT corresponding to the user that generated the new data.
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.

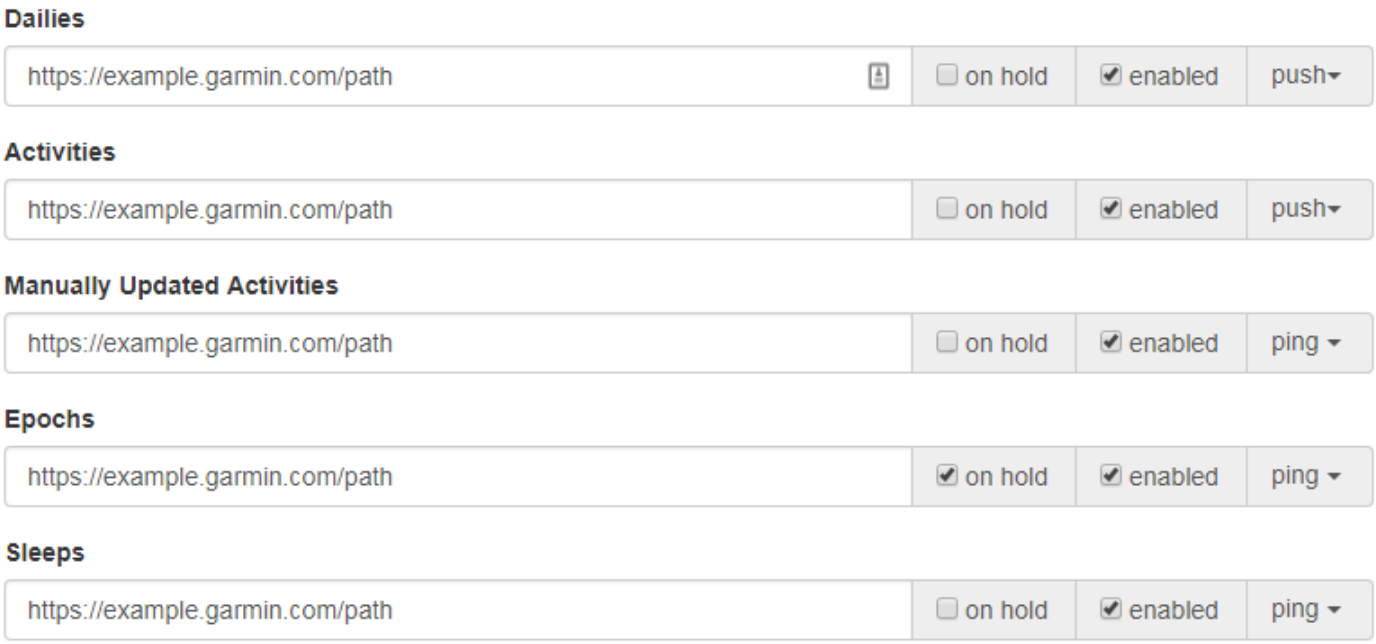
Example


```
{
   "epochs": [
      {
         "userId": "4aacafe82427c251df9c9592d0c06768",
         "userAccessToken": "8f57a6f1-26ba-4b05-a7cd-c6b525a4c7a2",
         "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
        }
    ]
}
```


5 Endpoint Configuration

Both the Push Service and the Ping Service can be configured using the Endpoint Configuration Tool found at https://healthapi.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.



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 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.

6 Health API Integration Tips

This section describes functionality that is important to understand when integrating with the Garmin 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 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 in to 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 in to 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 actually 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://healthapi.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 for the purposes of debugging or assisting 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.

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, use of this tool would be tedious in the event of a system-wide outage. The Health API support team (support@health.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 for the purposes of requesting 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.

6.3.6 User Authorization

This tool describes and performs the entire 3-legged OAuth process. It can be used to manually generate a User Access Token and authorize it for the currently used consumer key prior to any partner OAuth infrastructure being written.

6.3.7 Request Signing

This tool describes and demonstrates how to perform OAuth 1.0a request signing. Use of a third-party library is recommended, however manual signing can be useful for initial integration and debugging purposes. See the OAuth Specification document for more information on OAuth 1.0a request signing.

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.

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 in to 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.

Request

Resource URL
GET https://healthapi.garmin.com/wellness-api/rest/dailies

Request parameters

Parameter	Description
uploadStartTimeInSeconds	A UTC timestamp representing the beginning of the time range to search based on the moment the device actually uploaded the data. If this parameter is used it must be paired with uploadEndTimeInSeconds only. **Note:** This parameter corresponds to the value given in a Ping request.
uploadEndTimeInSeconds	A UTC timestamp representing the end of the time range to search based on the moment the device actually uploaded the data. If this parameter is used it must be paired with uploadStartTimeInSeconds only. This parameter corresponds to the value given in a Ping request. **Note:** This parameter corresponds to the value given in a Ping request.

Response

A successful response is a JSON array containing zero to many daily summaries. Please see Appendix E for possible error responses.

Each daily summary may contain the following fields:

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	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 backwards compatibility purposes. It can be ignored and will always default to WALKING.
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.
distanceInMeters	floating point	Distance traveled in meters.
activeTimeInSeconds	integer	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 through actual movement and activity during the monitoring period.
bmrKilocalories	integer	BMR Kilocalories burned by existing Basal Metabolic Rate (calculated based on user height/weight/age/other demographic data).
consumedCalories	integer	The number of calories that have been consumed by the user through food for that day (value subtracted from calorie goal). This value is received from MyFitnessPal and is not entered within Connect.
moderateIntensityDurationInSeconds	integer	Cumulative duration of activities of moderate intensity, lasting at least 600 seconds at a time. Moderate intensity is defined as activity with MET value range 3-6.
vigorousIntensityDurationInSeconds	integer	Cumulative duration of activities of vigorous intensity, lasting at least 600 seconds at a time. 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 of heart rate values 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 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.
stepsGoal	integer	The user's steps goal for this monitoring period.
netKilocaloriesGoal	integer	The user's goal for net caloric intake (consumed calories minus active calories) for this monitoring period. This field is related to integration with MyFitnessPal and may not be present for many users.

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

Request:
GET https://healthapi.garmin.com/wellness-api/rest/dailies?uploadStartTimeInSeconds=1452470400&uploadEndTimeInSeconds=1452556800

This request queries all daily summary records which were uploaded in the time between UTC timestamps *1452470400* (2016-01-11, 00:00:00 UTC) and *1452556800* (2016-01-12, 00:00:00 UTC).

Response:


```
[
      {
            "summaryId": " EXAMPLE_67891",
            "calendarDate": "2016-01-11",
            "activityType": "WALKING",
            "activeKilocalories": 321,
            "bmrKilocalories": 1731,
            "consumedCalories": 1121,
            "steps": 4210,
            "distanceInMeters": 3146.5,
            "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,
            "netKilocaloriesGoal": 2010,
            "intensityDurationGoalInSeconds": 1500,
            "floorsClimbedGoal": 18
      },
      {
            "summaryId": " EXAMPLE_67892",
            "activityType": "WALKING",
            "activeKilocalories": 304,
            "bmrKilocalories": 1225,
            "consumedCalories": 1926,
            "steps": 3305,
            "distanceInMeters": 2470.1,
            "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,
            "netKilocaloriesGoal": 2170,
            "intensityDurationGoalInSeconds": 1800,
            "floorsClimbedGoal": 20
      }
]
```


7.2 Third-Party Daily Summaries

This request is to retrieve a list of one or more daily summaries uploaded from third-party sources (e.g. Fitibit®) and not from Garmin devices. Third-Party Daily summaries must be explicitly and intentionally uploaded by the user, and third-party formats are often plain-text and user editable. Partners may choose whether or not to accept Third Party Dailies.

Note: Only the most recent third-party daily record is retained. If Garmin data exists for a given day, then the third-party data is ignored and will not be in the Health API. Garmin data and third-party data are never merged or intermixed.

Request

Resource URL
GET https://healthapi.garmin.com/wellness-api/rest/thirdPartyDailies

Request parameters

Parameter	Description
uploadStartTimeInSeconds	A UTC timestamp representing the beginning of the time range to search based on the moment the device actually uploaded the data. If this parameter is used it must be paired with uploadEndTimeInSeconds only. **Note:** This parameter corresponds to the value given in a Ping request.
uploadEndTimeInSeconds	A UTC timestamp representing the end of the time range to search based on the moment the device actually uploaded the data. If this parameter is used it must be paired with uploadStartTimeInSeconds only. This parameter corresponds to the value given in a Ping request. **Note:** This parameter corresponds to the value given in a Ping request.

Response

A successful response is a JSON array containing zero to many daily summaries. Please Appendix E for possible error responses.

Each daily summary may contain the following parameters:

Property	Type	Description
summaryId	string	Unique identifier for the summary.
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 backwards compatibility purposes. It can be ignored and will always default to WALKING.
durationInSeconds	integer	Length of the monitoring period in seconds.
steps	integer	Count of steps recorded during the monitoring period.
distanceInMeters	floating point	Distance traveled in meters.
activeTimeInSeconds	integer	Portion of the monitoring period (in seconds) in which the device wearer was active.
activeKilocalories	integer	Active kilocalories (dietary calories) burned through actual movement and activity during the monitoring period.
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, lasting at least 600 seconds at a time. Moderate intensity is defined as activity with MET value range 3-6.
vigorousIntensityDurationInSeconds	integer	Cumulative duration of activities of vigorous intensity, lasting at least 600 seconds at a time. 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 of heart rate values captured 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.
Source	string	The name of the source of the third party data. For example FITBIT.

Example

Request:

GET https://healthapi.garmin.com /wellness-api/rest/thirdPartyDailies?uploadStartTimeInSeconds=1472688000&uploadEndTimeInSeconds=1472774400

This request queries all daily summary records which were uploaded in the time between UTC timestamps 1472688000 (2016-09-01, 00:00:00 UTC) and1472774400 (2016-09-02, 00:00:00 UTC).

Response:


```
[ {
  "summaryId": "EXAMPLE_67891",
  "activityType": "WALKING",
  "activeKilocalories": 1136,
  "bmrKilocalories": 1736,
  "steps": 11467,
  "distanceInMeters": 14001.0,
  "durationInSeconds": 86400,
  "activeTimeInSeconds": 4680,
  "startTimeInSeconds": 1472688000,
  "startTimeOffsetInSeconds": 0,
  "floorsClimbed": 12,
  "source": "FITBIT"
}, {
  "summaryId": "EXAMPLE_67892",
  "activityType": "WALKING",
  "activeKilocalories": 1708,
  "bmrKilocalories": 1200,
  "steps": 13986,
```

```
  "distanceInMeters": 17091.0,
  "durationInSeconds": 86400,
  "activeTimeInSeconds": 8340,
  "startTimeInSeconds": 1472774400,
  "startTimeOffsetInSeconds": 0,
  "floorsClimbed": 42,
  "source": "FITBIT"
} ]
```


7.3 Activity Summaries

This request is to retrieve a list of one or more fitness activity summaries from the API.

Fitness activity summaries represent high-level information from discrete fitness activities, such as running or swimming, that are specifically and intentionally started by the user on their device. All wellness data, like steps and distance, contained in the Activity are already represented in the Daily summary and in the corresponding Epoch summaries, so Activity summaries should only be used for programs that wish to treat specific activity types in different ways, such as giving the user extra credit for going swimming three times in the same week.

For detailed activity information (e.g. heart rate, GPS track log, or other sensor information) see the Activity Details summary type.

Note:
Automatically detected Move IQ activities are not considered full-featured, discrete Activity Summaries. Move IQ events have their own summary type and may be configured and consumed separately (see below).

Request

Resource URL
GET https://healthapi.garmin.com/wellness-api/rest /activities

Request parameters

Parameter	Description
uploadStartTimeInSeconds	A UTC timestamp representing the beginning of the time range to search based on the moment the device actually uploaded the data. If this parameter is used it must be paired with uploadEndTimeInSeconds only. **Note:** This parameter corresponds to the value given in a Ping notification.
uploadEndTimeInSeconds	A UTC timestamp representing the end of the time range to search based on the moment the device actually uploaded the data. If this parameter is used it must be paired with uploadStartTimeInSeconds only. This parameter corresponds to the value given in a Ping request. **Note:** This parameter corresponds to the value given in a Ping notification.

Response

A successful response is a JSON array containing zero to many activity summaries. Please see Appendix E for possible error responses.

Each activity summary may contain the following parameters:

Property	Type	Description
summaryId	string	Unique identifier for the summary.
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	Text description of the activity type. See Appendix A for a complete list.
durationInSeconds	integer	Length of the monitoring period in seconds.
averageBikeCadenceInRoundsPerMinute	floating point	
averageHeartRateInBeatsPerMinute	integer	
averageRunCadenceInStepsPerMinute	floating point	
averageSpeedInMetersPerSecond	floating point	
averageSwimCadenceInStrokesPerMinute	floating point	
averagePaceInMinutesPerKilometer	floating point	
activeKilocalories	integer	
deviceName	string	Only Fitness Activities are associated with a specific Garmin device rather than the user's overall account. If a user wears two devices at once during the same time and starts a Fitness Activity on each then both will generate separate Activity summaries with two different deviceNames.
distanceInMeters	floating point	
maxBikeCadenceInRoundsPerMinute	floating point	
maxHeartRateInBeatsPerMinute	floating point	
maxPaceInMinutesPerKilometer	floating point	
maxRunCadenceInStepsPerMinute	floating point	
maxSpeedInMetersPerSecond	floating point	
numberOfActiveLengths	integer	
startingLatitudeInDegree	floating point	
startingLongitudeInDegree	floating point	
steps	integer	
totalElevationGainInMeters	floating point	
totalElevationLossInMeters	floating point	
isParent	boolean	If present and set to true, this activity is the parent activity of one or more child activities that should also be made available in the data feed to the partner. An activity of type MULTI_SPORT is an example of a parent activity.
parentSummaryId	integer	If present, this is the summaryId of the related parent activity. An activity of type CYCLING with a parent activity of type MULTI_SPORT is an example of this type of relationship.
manual	boolean	Indicates that the activity was manually entered directly on the Connect site. This property will only exist for manual activities.

Example

Request:
GET https://healthapi.garmin.com/wellness-api/rest/activities?uploadStartTimeInSeconds=1452470400&uploadEndTimeInSeconds=1452556800

This request queries all activity summary records which were uploaded in the time between UTC timestamps *1452470400* (2016-01-11, 00:00:00 UTC) and *1452556800* (2016-01-12, 00:00:00 UTC).

Response:


```
[
	{
		"summaryId": "EXAMPLE_12345",
		"activityType": "RUNNING",
		"startTimeInSeconds": 1452470400,
		"startTimeOffsetInSeconds": 0,
		"durationInSeconds": 11580,
		"averageSpeedInMetersPerSecond": 2.888999938964844,
		"distanceInMeters": 519818.125,
		"activeKilocalories": 448,
		"deviceName": "Forerunner 910XT",
		"averagePaceInMinutesPerKilometer": 0.5975272352046997
	},
	{
		"summaryId": "EXAMPLE_12346",
		"activityType": "CYCLING",
		"startTimeInSeconds": 1452506094,
		"startTimeOffsetInSeconds": 0,
		"durationInSeconds": 1824,
		"averageSpeedInMetersPerSecond": 8.75,
		"distanceInMeters": 4322.357,
		"activeKilocalories": 360,
		"deviceName": "Forerunner 910XT"
	}
]
```


7.4 Manually Updated Activity Summaries

Manual activities are created or edited by the user directly on the Connect site and not uploaded from a device. Manual activities can be identified by the property 'manual' = true and are separated out in to their own summary data type to allow greater control. This field will only exist if the activity is manually created or edited. Partners may choose to accept or ignore all or part of any manually created or updated Activities.

For testing purposes, activities can be uploaded or manually entered on Garmin Connect. The process to login and create activities is described below:

1. Login to https://connect.garmin.com (Create a user account if necessary)
2. Navigate to Activities -> All Activities -> + Manual Activity, or click here: https://connect.garmin.com/modern/activity/manual
3. Provide manual activity details and click Save.

Request

Resource URL
GET https://healthapi.garmin.com/wellness-api/rest/manuallyUpdatedActivities

Request parameters

Parameter	Description
uploadStartTimeInSeconds	A UTC timestamp representing the beginning of the time range to search based on the moment the user updated the data. If this parameter is used it must be paired with uploadEndTimeInSeconds only. **Note:** This parameter corresponds to the value given in a Ping notification.
uploadEndTimeInSeconds	A UTC timestamp representing the end of the time range to search based on the moment the user updated the data. If this parameter is used it must be paired with uploadStartTimeInSeconds only. This parameter corresponds to the value given in a Ping request. **Note:** This parameter corresponds to the value given in a Ping notification.

Response

A successful response is a JSON array containing zero to many activity summaries. Please see Appendix E for possible error responses.

Each activity summary may contain the following parameters:

Property	Type	Description
summaryId	string	Unique identifier for the summary.
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	Text description of the activity type. See Appendix A for a complete list.
durationInSeconds	integer	Length of the monitoring period in seconds.
averageBikeCadenceInRoundsPerMinute	floating point	
averageHeartRateInBeatsPerMinute	integer	
averageRunCadenceInStepsPerMinute	floating point	
averageSpeedInMetersPerSecond	floating point	
averageSwimCadenceInStrokesPerMinute	floating point	
averagePaceInMinutesPerKilometer	floating point	
activeKilocalories	integer	
deviceName	string	Always ‘unknown’ for manually created activities.
distanceInMeters	floating point	
maxBikeCadenceInRoundsPerMinute	floating point	
maxHeartRateInBeatsPerMinute	floating point	
maxPaceInMinutesPerKilometer	floating point	
maxRunCadenceInStepsPerMinute	floating point	

maxSpeedInMetersPerSecond	floating point	
numberOfActiveLengths	integer	
startingLatitudeInDegree	floating point	
startingLongitudeInDegree	floating point	
totalElevationGainInMeters	floating point	
totalElevationLossInMeters	floating point	
isParent	boolean	If present and set to true, this activity is the parent activity of one or more child activities that should also be made available in the data feed to the partner. An activity of type MULTI_SPORT is an example of a parent activity.
parentSummaryId	integer	If present, this is the summaryId of the related parent activity. An activity of type CYCLING with a parent activity of type MULTI_SPORT is an example of this type of relationship.
Manual	boolean	Indicates that the activity was manually updated directly on the Connect site.

Example

Request:

GET https://healthapi.garmin.com/wellness-api/rest/manuallyUpdatedActivities?uploadStartTimeInSeconds=1452470400&uploadEndTimeInSeconds=1452556800

This request queries all manually updated activity summary records which were uploaded in the time between UTC timestamps *1452470400* (2016-01-11, 00:00:00 UTC) and *1452556800* (2016-01-12, 00:00:00 UTC).

Response:


```
[
	{
		"summaryId": "EXAMPLE_12345",
		"activityType": "RUNNING",
		"startTimeInSeconds": 1452470400,
		"startTimeOffsetInSeconds": 0,
		"durationInSeconds": 11580,
		"averageSpeedInMetersPerSecond": 44.888999938964844,
		"distanceInMeters": 519818.125,
		"activeKilocalories": 448,
		"deviceName": "Forerunner 910XT",
		"averagePaceInMinutesPerKilometer": 0.5975272352046997,
		"manual": true
	},
	{
		"summaryId": "EXAMPLE_12346",
		"activityType": "CYCLING",
		"startTimeInSeconds": 1452506094,
		"startTimeOffsetInSeconds": 0,
		"durationInSeconds": 1824,
		"averageSpeedInMetersPerSecond": 8.75,
		"distanceInMeters": 4322.357,
		"activeKilocalories": 360,
		"deviceName": "Forerunner 910XT",
		"manual": true
```

```
        }
]
```

7.5 Activity Details Summaries

This request is to retrieve a list of one or more fitness activity details summaries from the API.

Fitness activity details summaries represent detailed information about discrete fitness activities, such as running or swimming, that are specifically and intentionally started by the user on their device. All wellness data, like steps and distance, contained in the activity are already represented in the Daily summary and in the corresponding Epoch summaries, so Activity Detail summaries should only be used for programs that wish to treat specific activity types in different ways, such as giving the user extra credit for going swimming three times in the same week.

Activity details summaries include all data recorded by the device as part of the Fitness Activity, including GPS coordinates and all recorded sensor data. **Activity details summaries are not available by default for newly-created consumer keys; to obtain access to this premium data please contact the Health API Support team (support@health.garmin.com).**

Request

Resource URL
GET https://healthapi.garmin.com/wellness-api/rest/activityDetails

Request parameters

Parameter	Description
uploadStartTimeInSeconds	A UTC timestamp representing the beginning of the time range to search based on the moment the device actually uploaded the data. If this parameter is used it must be paired with uploadEndTimeInSeconds only. **Note:** This parameter corresponds to the value given in a Ping notification.
uploadEndTimeInSeconds	A UTC timestamp representing the end of the time range to search based on the moment the device actually uploaded the data. If this parameter is used it must be paired with uploadStartTimeInSeconds only. This parameter corresponds to the value given in a Ping request. **Note:** This parameter corresponds to the value given in a Ping notification.

Response

A successful response is a JSON array containing zero to many activity detail summaries. Each activity detail contains an activity summary and an optional list of samples. The samples list will be empty if the activity is manual or details are not supported by the device. Samples may be as frequent as once per second, and values should be considered valid until the next sample.

Each activity detail contains a summary field that may contain the following parameters:

Property	Type	Description
summaryId	string	Unique identifier for the summary.
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	Text description of the activity type. See Appendix A for a complete list.
durationInSeconds	integer	Length of the monitoring period in seconds.
averageBikeCadenceInRoundsPerMinute	floating point	
averageHeartRateInBeatsPerMinute	integer	
averageRunCadenceInStepsPerMinute	floating point	
averageSpeedInMetersPerSecond	floating point	
averageSwimCadenceInStrokesPerMinute	floating point	
averagePaceInMinutesPerKilometer	floating point	
activeKilocalories	integer	
deviceName	string	Only Fitness activities are associated with a specific Garmin device rather than the user's overall account. If the user wears two devices at once at the same time and starts a Fitness Activity on each then both will generate separate Activities with two different deviceNames.
distanceInMeters	floating point	
maxBikeCadenceInRoundsPerMinute	floating point	
maxHeartRateInBeatsPerMinute	floating point	
maxPaceInMinutesPerKilometer	floating point	
maxRunCadenceInStepsPerMinute	floating point	
maxSpeedInMetersPerSecond	floating point	
numberOfActiveLengths	integer	
startingLatitudeInDegree	floating point	
startingLongitudeInDegree	floating point	
steps	integer	
totalElevationGainInMeters	floating point	
totalElevationLossInMeters	floating point	
isParent	boolean	If present and set to true, this activity is the parent activity of one or more child activities that should also be made available in the data feed to the partner. An activity of type MULTI_SPORT is an example of a parent activity.
parentSummaryId	integer	If present, this is the summaryId of the related parent activity. An activity of type CYCLING with a parent activity of type MULTI_SPORT is an example of this type of relationship.
manual	boolean	Indicates that the activity was manually entered directly on the Connect site. This property will only exist for manual activities.

Each activity detail may contain a list of samples, each of which may containing the following:

Property	Type	Description
startTimeInSeconds	integer	Start time of the sample in seconds since January 1, 1970, 00:00:00 UTC (Unix timestamp).
latitudeInDegree	floating point	Latitude in decimal degrees (DD)
longitudeInDegree	floating point	Longitude in decimal degrees (DD)
elevationInMeters	floating point	
airTemperatureCelcius	floating point	
heartrate	Integer	Heart rate in beats per minute

speedMetersPerSecond	floating point	
stepsPerMinute	floating point	
totalDistanceInMeters	floating point	
timerDurationInSeconds	integer	The amount of "timer time" in an activity
clockDurationInSeconds	integer	The amount of real-world "clock time" from the start of an activity to the end
movingDurationInSeconds	integer	The amount of "timer time" during which the athlete was moving (above a threshold speed)
powerInWatts	floating point	The amount of power expended in watts
bikeCadenceInRPM	floating point	Cycling cadence in revolutions per minute
swimCadenceInStrokesPerMinute	floating point	Swim cadence in strokes per minute

Tip: In all cases, movingDurationInSeconds <= timerDurationInSeconds <= clockDurationInSeconds.

For example, a user is going for a run. He starts the timer at exactly noon. At 12:30 he pauses the timer (either manually or using auto-pause) to stop and chat with a friend, and at 12:35 he resumes the timer. At 12:40 he stands still for 2 minutes, waiting on a traffic signal at a busy intersection, then finishes his run and manually stops the timer at 1:00 pm.

clockDurationInSeconds = 60 minutes (12:00 - 1:00)
timerDurationInSeconds = 55 minutes (12:00-12:30 + 12:35-1:00)
movingDurationInSeconds = 53 minutes (12:00-12:30 + 12:35-12:40 + 12:42-1:00)

Activity Details records may also contain lap data indicating when the user initiated a new lap, either manually or by Auto Lap functionality (https://www8.garmin.com/manuals/webhelp/vivoactive3/EN-US/GUID-97010D91-30E5-42CD-871D-ED17CA77C5AC.html). Each lap object contains the following:

Property	Type	Description
startTimeInSeconds	integer	Start time of the lap in seconds since January 1, 1970, 00:00:00 UTC (Unix timestamp).

Request:

GET https://healthapi.garmin.com/wellness-api/rest/activityDetails?uploadStartTimeInSeconds=1452470400&uploadEndTimeInSeconds=1452556800

This request queries all activity details summary records which were uploaded in the time between UTC timestamps *1452470400* (2016-01-11, 00:00:00 UTC) and *1452556800* (2016-01-12, 00:00:00 UTC).

Response:


```
[
   {
      "summaryId": "14311627-detail",
      "summary": {
         "durationInSeconds": 1789,
         "startTimeInSeconds": 1512234126,
         "startTimeOffsetInSeconds": -25200,
         "activityType": "RUNNING",
         "averageHeartRateInBeatsPerMinute": 144,
         "averageRunCadenceInStepsPerMinute": 84.0,
```

```
        "averageSpeedInMetersPerSecond": 2.781,
        "averagePaceInMinutesPerKilometer": 15.521924,
        "activeKilocalories": 367,
        "deviceName": "forerunner935",
        "distanceInMeters": 4976.83,
        "maxHeartRateInBeatsPerMinute": 159,
        "maxPaceInMinutesPerKilometer": 10.396549,
        "maxRunCadenceInStepsPerMinute": 106.0,
        "maxSpeedInMetersPerSecond": 4.152,
        "startingLatitudeInDegree": 51.053232522681355,
        "startingLongitudeInDegree": -114.06880217604339,
        "steps": 5022,
        "totalElevationGainInMeters": 16.0,
        "totalElevationLossInMeters": 22.0
    },
    "samples": [
        {
            "startTimeInSeconds": 1512234126,
            "latitudeInDegree": 51.053232522681355,
            "longitudeInDegree": -114.06880217604339,
            "elevationInMeters": 1049.4000244140625,
            "airTemperatureCelcius": 28.0,
            "heartRate": 90,
            "speedMetersPerSecond": 0.0,
            "stepsPerMinute": 57.0,
            "totalDistanceInMeters": 0.17000000178813934,
            "timerDurationInSeconds": 0,
            "clockDurationInSeconds": 0,
            "movingDurationInSeconds": 0
        },
        {
            "startTimeInSeconds": 1512234127,
            "latitudeInDegree": 51.05323604308069,
            "longitudeInDegree": -114.06880334950984,
            "elevationInMeters": 1049.0,
            "airTemperatureCelcius": 28.0,
            "heartRate": 92,
            "speedMetersPerSecond": 0.0,
            "stepsPerMinute": 58.0,
            "totalDistanceInMeters": 0.5699999928474426,
            "timerDurationInSeconds": 1,
            "clockDurationInSeconds": 1,
            "movingDurationInSeconds": 0
        }
    ],
    "laps": [
        {
            "startTimeInSeconds": 1512234126
        },
        {
            "startTimeInSeconds": 1512234915
        }
    ]
  }
]
```

7.6 Activity Files

Activity details are also available as raw FIT, TCX, and GPX files. These are the actual files recorded by the wearable as part of the Fitness Activity, including GPS coordinates, all recorded sensor data, and any product-specific data that may not be exposed as part of the parsed Activity Details Summary. Raw files are not available by default for newly-created consumer keys; to obtain access to this premium data please contact the Health API Support team (support@health.garmin.com).

Parsing of raw files is the responsibility of the partner. When deciding between Activity Details Summaries and Activity Files it is generally recommended to only choose Files if there are specific required fields or details in the Files that are not available in the Summaries. The recommend publicly available parsers and schemas are:

- TCX: https://www8.garmin.com/xmlschemas/TrainingCenterDatabasev2.xsd
- GPX: https://www.topografix.com/gpx.asp
- FIT: https://www.thisisant.com/resources/fit

Unlike normal Summaries, Activity Files are not available as a Push integration. Files are only available in response to a Ping by calling the specified callbackURL. The Ping's body is JSON formatted as follows:

```
"activityFiles" : [ {
    "userId" : "3b9d232ce7782a5c3fb036d09956c5de",
    "userAccessToken" : "9dc2ef1a-cd20-42e5-b240-b2dc747ed330",
    "fileType" : "FIT",
    "callbackURL" : "https://healthapi.garmin.com/wellness-
api/rest/activityFile?id=123456"
     },
     {
     "userId" : "3b9d232ce7782a5c3fb036d09956c5de",
     "userAccessToken" : "9dc2ef1a-cd20-42e5-b240-b2dc747ed330",
     "fileType" : "FIT",
     "callbackURL" : "https://healthapi.garmin.com/wellness-
     api/rest/activityFile?id=654321"
}
]
```

Unlike a normal Ping body, the file type (TCX, GPX, or FIT) is specified in the filetype field and the callback URL specifies the Activity File by an ID rather than by the upload time range.

7.7 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.

Request

Resource URL
GET https://healthapi.garmin.com/wellness-api/rest/epochs
Request parameters

Parameter	Description
uploadStartTimeInSeconds	A UTC timestamp representing the beginning of the time range to search based on the moment the device actually uploaded the data. If this parameter is used it must be paired with uploadEndTimeInSeconds only. **Note:** This parameter corresponds to the value given in a Ping notification.
uploadEndTimeInSeconds	A UTC timestamp representing the end of the time range to search based on the moment the device actually uploaded the data. If this parameter is used it must be paired with uploadStartTimeInSeconds only. This parameter corresponds to the value given in a Ping request. **Note:** This parameter corresponds to the value given in a Ping notification.

Response

A successful response is a JSON array containing zero to many wellness monitoring summaries. Please see Appendix E for possible error responses.

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
distanceInMeters	floating point	Distance traveled in meters
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

Request:
GET https://healthapi.garmin.com/wellness-api/rest/epochs?uploadStartTimeInSeconds=1454418900&uploadEndTimeInSeconds=1454419800

This request queries all wellness monitoring summary records which were uploaded in the time between UTC timestamps *1452470400* (2016-01-11, 00:00:00 UTC) and *1452556800* (2016-01-12, 00:00:00 UTC).

Response:


```
[
      {
            "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
      }
]
```


7.8 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 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 labelled 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 it is possible for two different Sleep summaries to begin on the same day if, for example, the user goes to bed after midnight, wakes up, and then goes to bed prior to 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 devices is set as the preferred activity tracker and is actually 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.

Request

Resource URL
GET
https://healthapi.garmin.com/wellness-api/rest/sleeps

Request parameters

Parameter	Description
uploadStartTimeInSeconds	A UTC timestamp representing the beginning of the time range to search based on the moment the device actually uploaded the data. If this parameter is used it must be paired with uploadEndTimeInSeconds only. **Note:** This parameter corresponds to the value given in a Ping notification.
uploadEndTimeInSeconds	A UTC timestamp representing the end of the time range to search based on the moment the device actually uploaded the data. If this parameter is used it must be paired with uploadStartTimeInSeconds only. This parameter corresponds to the value given in a Ping request. **Note:** This parameter corresponds to the value given in a Ping notification.

Response

A successful response is a JSON array containing zero to many sleep summaries. Please see Appendix E for possible error responses.

Each sleep 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	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 monitoring period in seconds.
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	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. AUTO_TENTATIVE: The sleep start and stop times were auto-detected by Garmin Connect using accelerometer data. However, it is possible that further refinements to this sleep record will come later. This could be because the user is still asleep or could be 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.

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.

Example

Request:
*GET https://healthapi.garmin.com/wellness-api/rest/sleeps?uploadStartTimeInSeconds=*1467823680*&uploadEndTimeInSeconds=*1467860580

This request queries all sleep summary records which were uploaded in the time between UTC timestamps 1467823680 (2016-07-06, 16:48:00 UTC) and 1467860580 (2016-07-07, 03:03:00 UTC).

Response:


```
[
      {
            "summaryId": "EXAMPLE_567890",
            "calendarDate": "2016-01-10",
            "durationInSeconds": 15264,
            "startTimeInSeconds": 1452419581,
            "startTimeOffsetInSeconds": 7200,
            "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
                        }
                  ],
                  "rem": [
                        {
                              "startTimeInSeconds": 1452476940,
                              "endTimeInSeconds": 1452479082
                        }
                  ]
            },
            "validation": "DEVICE",
            "timeOffsetSleepSpo2": {
                  "0": 95,
                  "60": 96,
                  "120": 97,
                  "180": 93,
                  "240": 94,
                  "300": 95,
                  "360": 96
            }
      }
]
```

7.9 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 three ways. Users can manually enter their weight on Garmin Connect. This results in a summary with only time and weight.

Users may also connect their MyFitnessPal account to their Garmin Connect account and update their weight on MyFitnessPal. This results in a summary that also just has a 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.

Request

Resource URL
GET
https://healthapi.garmin.com/wellness-api/rest/bodyComps

Request parameters

Parameter	Description
uploadStartTimeInSeconds	A UTC timestamp representing the beginning of the time range to search based on the moment the device actually uploaded the data. If this parameter is used it must be paired with uploadEndTimeInSeconds only. **Note:** This parameter corresponds to the value given in a Ping notification.
uploadEndTimeInSeconds	A UTC timestamp representing the end of the time range to search based on the moment the device actually uploaded the data. If this parameter is used it must be paired with uploadStartTimeInSeconds only. This parameter corresponds to the value given in a Ping request. **Note:** This parameter corresponds to the value given in a Ping notification.

Response

A successful response is a JSON array containing zero to many body composition summaries. Please see Appendix E for possible error responses.

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

Request:
GET https://healthapi.garmin.com/wellness-api/rest/bodyComps?uploadStartTimeInSeconds=1452470400&uploadEndTimeInSeconds=1452556800

This request queries all body composition summary records which were uploaded in the time between UTC timestamps *1452470400* (2016-01-11, 00:00:00 UTC) and *1452556800* (2016-01-12, 00:00:00 UTC).

Response:


```
[
      {
            "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
      }

]
```


7.10 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.

Request

Resource URL

GET
https://healthapi.garmin.com/wellness-api/rest/stressDetails

Request parameters

Parameter	Description
uploadStartTimeInSeconds	A UTC timestamp representing the beginning of the time range to search based on the moment the device actually uploaded the data. If this parameter is used it must be paired with uploadEndTimeInSeconds only. **Note:** This parameter corresponds to the value given in a Ping notification.
uploadEndTimeInSeconds	A UTC timestamp representing the end of the time range to search based on the moment the device actually uploaded the data. If this parameter is used it must be paired with uploadStartTimeInSeconds only. This parameter corresponds to the value given in a Ping request. **Note:** This parameter corresponds to the value given in a Ping notification.

Response

A successful response is a JSON array containing zero to many stress details summaries. Please see Appendix E for possible error responses.

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 this summary would 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.
timeOffsetBodyBatteryDetails	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 .

Example

Request:
GET https://healthapi.garmin.com/wellness-api/rest/
stressDetails?uploadStartTimeInSeconds=1490372394&uploadEndTimeInSeconds=1490372634

This request queries all stress details summary records which were uploaded in the time between UTC timestamps 1490372394 (2017-03-24, 16:19:54 UTC) and 1490372634 (2017-03-24, 16:23:54 UTC).

Response:


```
[
      {
            "summaryId": " EXAMPLE_6789124",
            "calendarDate": "2017-03-23",
            "startTimeInSeconds": 1490245200,
            "startTimeOffsetInSeconds": 0,
            "durationInSeconds": 540,
            "timeOffsetStressLevelValues": {
                  "0": 18,
                  "180": 51,
                  "360": 28,
                  "540": 29
            },
            "timeOffsetBodyBatteryDetails": {
                        "0": 55,
                        "180": 56,
                        "360": 59
            }
]
```


7.11 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.

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.

Request

Resource URL
GET
https://healthapi.garmin.com/wellness-api/rest/userMetrics

Request parameters

Parameter	Description
uploadStartTimeInSeconds	A UTC timestamp representing the beginning of the time range to search based on the moment the device actually uploaded the data. If this parameter is used it must be paired with uploadEndTimeInSeconds only. **Note:** This parameter corresponds to the value given in a Ping notification.
uploadEndTimeInSeconds	A UTC timestamp representing the end of the time range to search based on the moment the device actually uploaded the data. If this parameter is used it must be paired with uploadStartTimeInSeconds only. This parameter corresponds to the value given in a Ping request. **Note:** This parameter corresponds to the value given in a Ping notification.

Response

A successful response is a JSON array containing zero to many user metrics summaries. Please see Appendix E for possible error responses.

Each user metrics 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'.
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.
fitnessAge	integer	An estimation of the 'age' of the user's fitness level, 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.

Example

Request:
GET https://healthapi.garmin.com/wellness-api/rest/userMetrics?uploadStartTimeInSeconds=1490372394&uploadEndTimeInSeconds=1490372634

This request queries all user metrics records which were uploaded in the time between UTC timestamps 1490372394 (2017-03-24, 16:19:54 UTC) and 1490372634 (2017-03-24, 16:23:54 UTC).

Response:

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

7.12 Move IQ Summaries

Move IQ Event summaries are a feed of activities which have been automatically detected by the device based on movement patterns, like running or biking. These are not activities initiated by the user. Please note that wellness data, like steps and distance, from Move IQ events are already included in the Daily and Epoch summaries.

Due to their automatically-detected nature, Move IQ events are not considered a fitness activity, do not contain the same details as activities, and cannot be edited by the user with Garmin Connect. These events should be considered a labeled-timespan on top of normal Daily or Epoch summary details, matching their representation within Garmin Connect.

For more feature-level information on Move IQ events, please see: https://support.garmin.com/faqSearch/en-US/faq/content/zgFpy8MShkArqAxCug5wC6 . Move IQ activities are also known as Automatic Activity Detection in older devices or documentation.

Request

Resource URL
GET
https://healthapi.garmin.com/wellness-api/rest/moveiq

Request parameters

Parameter	Description
uploadStartTimeInSeconds	A UTC timestamp representing the beginning of the time range to search based on the moment the device actually uploaded the data. If this parameter is used it must be paired with uploadEndTimeInSeconds only. **Note:** This parameter corresponds to the value given in a Ping notification.
uploadEndTimeInSeconds	A UTC timestamp representing the end of the time range to search based on the moment the device actually uploaded the data. If this parameter is used it must be paired with uploadStartTimeInSeconds only. **Note:** This parameter corresponds to the value given in a Ping notification.

Response

A successful response is a JSON array containing zero to many Move IQ event summaries. Please see Appendix E for possible error responses.

Each Move IQ event 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).
offsetInSeconds	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.
activityType	string	The activity type that has been identified for this timespan.
activitySubType	string	The activity subtype that has been identified for this timespan.

Example

Request:
GET https://healthapi.garmin.com/wellness-api/rest/moveiq?uploadStartTimeInSeconds=1490372394&uploadEndTimeInSeconds=1490372634

This request queries all Move IQ event records which were uploaded in the time between UTC timestamps 1490372394 (2017-03-24, 16:19:54 UTC) and 1490372634 (2017-03-24, 16:23:54 UTC).

Response:

```
[
  {
    "summaryId": " EXAMPLE_843244",
    "calendarDate": "2017-03-23",
    "startTimeInSeconds": 1490245200,
    "durationInSeconds": 738,
    "offsetInSeconds": 0,
    "activityType": "Running",
    "activitySubType": "Hurdles"
   }
]
```

7.13 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 not present or set to false, the timeOffsetSpo2Values map contains an 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.

Backfill is supported for Pulse Ox summaries, but only for Acclimation data.

Request

Resource URL
GET
https://healthapi.garmin.com/wellness-api/rest/pulseOx

Request parameters

Parameter	Description
uploadStartTimeInSeconds	A UTC timestamp representing the beginning of the time range to search based on the moment the device actually uploaded the data. If this parameter is used it must be paired with uploadEndTimeInSeconds only. **Note:** This parameter corresponds to the value given in a Ping notification.
uploadEndTimeInSeconds	A UTC timestamp representing the end of the time range to search based on the moment the device actually uploaded the data. If this parameter is used it must be paired with uploadStartTimeInSeconds only. **Note:** This parameter corresponds to the value given in a Ping notification.

Response

A successful response is a JSON array containing zero to many Pulse Ox summaries. Please see Appendix E for possible error responses.

Each Pulse Ox 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).
offsetInSeconds	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.
onDemand	boolean	A Boolean to show whether this pulse ox summary represents an on-demand reading or an averaged acclimation reading.

Example

Request:

GET https://healthapi.garmin.com/wellness-api/rest/pulseOx?uploadStartTimeInSeconds=1535300706&uploadEndTimeInSeconds=1535410706

This request queries all Pulse Ox records which were uploaded in the time between UTC timestamps 1535300706 and 1535410706.

Response:

```
[
  {
    "summaryId": " EXAMPLE_34343",
    "calendarDate": "2018-08-27",
    "startTimeInSeconds": 1535400706,
    "durationInSeconds": 86400,
    "offsetInSeconds": 7200,
    "timeOffsetSpo2Values": {
        "0": 98,
        "3600": 95,
        "7200": 93,
        "10800": 98
    }
}
]
```

7.14 Menstrual Cycle Tracking (MCT) Summaries

The Menstrual Cycle Tracking feature (https://connect.garmin.com/features/menstrual-cycle-tracking/) available on some Garmin devices allows users to track information about their cycle schedule and log symptoms. The MCT Summary only returns information related to cycle schedule (see response parameters) and does not make the information about symptoms available.

In addition to providing consent to share their Garmin Connect data with your Health API program, users must also indicate their permission to share their MCT schedule information.

Access to configure the MCT summary endpoints is not provided by default to partners. Please contact support@health.garmin.com to request access and provide an explanation of the feature you intend to build based on this data.

Note: Backfill is not supported for MCT summaries.

Request

Resource URL

GET https://healthapi.garmin.com/wellness-api/rest/mct

Request parameters

Parameter	Description
uploadStartTimeInSeconds	A UTC timestamp representing the beginning of the time range to search based on the moment, a user logs their symptoms. **Note:** This parameter corresponds to the value given in a Ping notification.
uploadEndTimeInSeconds	A UTC timestamp representing the end of the time range to search based on the moment, a user logs their symptoms. **Note:** This parameter corresponds to the value given in a Ping notification.

Response

A successful response is a JSON array containing zero to one MCT summaries. Please see Appendix E for possible error responses.

Each MCT summary may contain the following parameters:

Property	Type	Description
summaryId	string	Unique identifier for the summary
periodStartDate	string	The calendar date representing period start date. The date format is 'yyyy-mm-dd'
dayInCycle	integer	Represents ^{nth} day in cycle
periodLength	integer	Number of days indicating how long a period usually last
currentPhase	integer	Indicates the phase in this cycle like menstruation, fertile, etc.,

lengthOfCurrentPhase	integer	Represents the length of current phase in days
daysUntilNextPhase	integer	Number of days remaining to reach the next predicted phase
predictedCycleLength	integer	Number of days predicted to be the current cycle length
isPredictedCycle	boolean	A Boolean to show if this summary is a predicted cycle or not
cycleLength	integer	A user logged cycle length
lastUpdatedTimeInSeconds	integer	Time in seconds showing when a user logged their symptoms

Example

Request:

GET https://healthapi.garmin.com/wellness-api/rest/mct?uploadStartTimeInSeconds=1567747138&uploadEndTimeInSeconds=1567804738

This request queries the MCT summaries which a user has logged their symptoms in the time between UTC timestamps 1567747138 and 1567804738.

Response:


```
[
    {
        "summaryId": "x2faf44e-16cfcc8d020",
        "periodStartDate": "2019-09-03",
        "dayInCycle": 1,
        "periodLength": 1,
        "currentPhase": 1,
        "lengthOfCurrentPhase": 1,
        "daysUntilNextPhase": 1,
        "predictedCycleLength": 16,
        "isPredictedCycle": false,
        "cycleLength": 16,
        "lastUpdatedTimeInSeconds": 1567609114
    }
]
```


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 prior to 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 time 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.

Request

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

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

Resource URL for activity summaries, details, and files
GET https://healthapi.garmin.com/wellness-api/rest/backfill/activities

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

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

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

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

Resource URL for Move IQ event summaries
GET https://healthapi.garmin.com/wellness-api/rest/backfill/moveiq

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

Request parameters

Parameter	Description
summaryStartTimeInSeconds	A UTC timestamp representing 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 representing 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://healthapi.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 User Endpoints

Unlike Summary endpoints which fetch user data, User Endpoints perform operations on the user's account itself. The availability and scope of the operations are intentionally limited to protect the user's privacy.

9.1 Delete User Access Token

This service provides the ability to remove a user from your program, specific to the consumer key being used, by deleting the UAT. After being called, a final User Deregistration notification will be sent as though the user had withdrawn access through Garmin Connect (if enabled).

Immediately following the Deregistration ping, all notifications for that user will immediately stop and any attempts to request data with that UAT will be rejected as unauthorized. The deleted UAT cannot be restored. The same user (with the same Garmin Connect account) going through the OAuth a second time will generate a completely different UAT.

This endpoint must be called if the partner website or application provides a "Delete My Account" or "Opt-Out" mechanism outside of the normal Garmin Connect consent removal process or in any other case where the user would reasonably believe the partner program is giving them the opportunity to remove their consent to share Garmin data.

Request URL to delete a user registration
DELETE https://healthapi.garmin.com/wellness-api/rest/user/registration

No parameters are required for this request. The user access token is taken from the OAuth header.

Response:

On a successful request, this service returns HTTP 204 (no content) with no response body. Please see Appendix E for possible error responses.

9.2 Get Health API User ID

Each Garmin Connect user has a unique Health API ID associated with them that will persist across multiple UATs. For instance, if a user deletes their association through Garmin Connect and then, later, completes the OAuth process to generate a new User Access Token with the same Garmin Connect account, the second token will still have the same Health API User ID as the first token. Similarly, if a partner is managing multiple programs and the user signs up for each of them, the Health API User ID returned for each of the UATs will match.

The Health API ID provides no identifying information and is not used in any other Garmin API, web service, or system. There is no reason to ever pass the Health API User ID back to the Health API as user lookup will always be performed using the User Access Token in the Authorization header.

Request URL to fetch a Health API User ID
GET https://healthapi.garmin.com/wellness-api/rest/user/id

No parameters are required for this request.

Response:

```
{
   "userId": "d3315b1072421d0dd7c8f6b8e1de4df8"
}
```

Appendix A – Activity Types

Below is the list of valid activity types referenced in Garmin Connect fitness activity summaries.

Activity Type
ALL
UNCATEGORIZED
SEDENTARY
SLEEP
RUNNING
STREET_RUNNING
TRACK_RUNNING
TRAIL_RUNNING
TREADMILL_RUNNING
CYCLING
CYCLOCROSS
DOWNHILL_BIKING
INDOOR_CYCLING
MOUNTAIN_BIKING
RECUMBENT_CYCLING
ROAD_BIKING
TRACK_CYCLING
FITNESS_EQUIPMENT
ELLIPTICAL
INDOOR_CARDIO
INDOOR_ROWING
STAIR_CLIMBING
STRENGTH_TRAINING
HIKING
SWIMMING
LAP_SWIMMING
OPEN_WATER_SWIMMING
WALKING
CASUAL_WALKING
SPEED_WALKING
TRANSITION
SWIMTOBIKETRANSITION

BIKETORUNTRANSITION
RUNTOBIKETRANSITION
MOTORCYCLING
OTHER
BACKCOUNTRY_SKIING_SNOWBOARDING
BOATING
CROSS_COUNTRY_SKIING
DRIVING_GENERAL
FLYING
GOLF
HORSEBACK_RIDING
INLINE_SKATING
MOUNTAINEERING
PADDLING
RESORT_SKIING_SNOWBOARDING
ROWING
SAILING
SKATE_SKIING
SKATING
SNOWMOBILING
SNOW_SHOE
STAND_UP_PADDLEBOARDING
WHITEWATER_RAFTING_KAYAKING
WIND_KITE_SURFING

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 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 in the creation of other metrics. Motion Intensity is calculated at minute-level granularity as a number between 0 and 7, with 0 being absolutely still and 7 being constant, sharp motion. Unlike steps, distance, or activity type, which take net movement in to 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 absolutely 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 error message in the response body for details.
401 - Unauthorized	The authorization for the request failed. See 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.
500 - Internal Server Error	Any server error that does not fall in to one of the above categories.

Example

Request:

```
GET https://healthapi.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."
}
```