



# Fitbit Web API Data Dictionary

Version 8; last updated Aug 22, 2023

Fitbit's free public Web APIs are designed to give developers the ability to quickly and easily create applications to leverage authorized Fitbit user data. Organizations planning to utilize Fitbit's APIs will need computing resources along with developers who have technical knowledge of API architecture. The design of your application and the endpoints (data) needed will largely be defined by the purpose of your application. If you're a researcher designing an application to collect Fitbit user data for a study, you'll want to assess which endpoints are relevant to your study. Additionally, your application can combine endpoints to create new metrics. In other words, there are an almost endless number of ways to collect, visualize and combine Fitbit data utilizing our web APIs. It's important to remember to account for data storage costs - especially when collecting intraday data.

In this document, we've defined our most commonly used and relevant endpoints related to activity, heart rate, sleep, body, food logging and devices to help you design your study and/or application. For a complete list of endpoints and their response, please see the endpoint specific pages.

Many of the endpoints return data in the user defined or applications localization settings. The data that can be localized is listed in our [Application Design](#) documentation.

Lastly, please note that functionality of Fitbit devices varies. Refer to our current [product page](#) for details on all our current products or if you'd like to speak with a member of our sales team, please contact us [here](#).

## Activity

The following information is activity data logged by Fitbit devices worn by Fitbit users, activity data logged by Fitbit users through the Fitbit mobile application, or 3rd party application activity data.

### GET Activity

Element Name	Datatype	Description
activityCalories	integer	The number of calories burned for the day during periods the user was active above <a href="#">sedentary</a> level. This does not include calories burned from Basal



		metabolic rate ( <a href="#">BMR</a> ). Endpoint(s): <a href="#">Get Daily Activity Summary</a>
activeDuration	integer	The amount of time in each activityLevel. Time in milliseconds. Endpoint(s): <a href="#">Get Activity Logs List</a>
activeMinutes	integer	User defined goal for daily active minutes. Endpoint(s): <a href="#">Get Daily Activity Summary</a>
activeZoneMinutes	Integer	Daily or weekly active zone minutes goal. Endpoint(s): <a href="#">Get Activity Goals</a>
activityId	integer	The recorded exercise's identifier number. For example, the activityId for "Run" is 90009. For a complete list of exercise identifiers, see <a href="#">Browse Activity Type</a> endpoint. Endpoint(s): <a href="#">Get Daily Activity Summary</a>
activityLevel	integer	Returns minutes spent in each activity zone: <a href="#">sedentary, lightly active, moderately active, and very active</a> . Endpoint(s): <a href="#">Get Activity Logs List</a>
activityName	string	Name of the recorded exercise (i.e. Walk, Run, Elliptical, Hike, and custom exercises) Endpoint(s): <a href="#">Get Activity Logs List</a>
altitudemeters	integer	The altitude meters recorded at the specified time. Endpoint(s): <a href="#">Get Activity TCX</a>
averageHeartRate	integer	Average heart rate during the exercise from start to finish. Endpoint(s): <a href="#">Get Activity Logs List</a>
calories	integer	The number of calories burned associated with the activity, goal, or summary total. The value returned is a minute by minute summary total of activity minutes



		<p>recorded above <a href="#">sedentary</a> level, excluding caloriesBMR.</p> <p>Endpoint(s): <a href="#">Get Daily Activity Summary</a>, <a href="#">Get Frequent Activities</a>, <a href="#">Get Recent Activity Types</a></p>
calories	integer	<p>Number of calories burned during the user's exercise-defined lap</p> <p>Endpoint(s): <a href="#">Get Activity TCX</a></p>
caloriesBMR	integer	<p>Total number of <a href="#">BMR</a> calories burned for the day. Does not include calories burned above <a href="#">sedentary</a> level.</p> <p>Endpoint(s): <a href="#">Get Daily Activity Summary</a></p>
caloriesEstimationMu	integer	<p>Total estimated calories burned for the day based on <a href="#">measurement uncertainty</a>.</p> <p>Endpoint(s): <a href="#">Get Daily Activity Summary</a></p>
caloriesLink	string	<p>Web API endpoint to call to get the specific calories burned for the named exercise.</p> <p>Endpoint(s): <a href="#">Get Activity Logs List</a></p>
caloriesOut	integer	<p>Total calories burned associated with the activity, goal, summary totals. Inclusive of activityCalories and caloriesBMR.</p> <p>Endpoint(s): <a href="#">Get Daily Activity Summary</a>, <a href="#">Get Activity Goals</a></p>
caloriesOutUnestimated	integer	<p>Total unestimated calories burned for the day.</p> <p>Endpoint(s): <a href="#">Get Daily Activity Summary</a></p>
date	date	<p>Date of log; in the format <b>YYYY-MM-DD</b>.</p> <p>Endpoint(s): <a href="#">Get Lifetime Stats</a></p>
distance	integer	<p>Distance traveled associated with the recorded activity, goal, summary, and lifetime totals.</p> <p>Endpoint(s): <a href="#">Get Daily Activity Summary</a>, <a href="#">Get Frequent Activities</a>, <a href="#">Get Recent Activity Types</a>, <a href="#">Get</a></p>



		<a href="#">Activity Goals</a> , <a href="#">Get Lifetime Stats</a>
distancemeters	integer	Number of meters traveled during the exercise lap Endpoint(s): <a href="#">Get Activity TCX</a>
distanceUnit	string	Distance units defined by the Accept-Language header. Endpoint(s): <a href="#">Get Activity Logs List</a>
distances	list	Total distance accumulated for the day for the following elements: <ul style="list-style-type: none"> <li>• total</li> <li>• tracker</li> <li>• <a href="#">loggedActivities</a></li> <li>• <a href="#">veryActive</a></li> <li>• <a href="#">moderatelyActive</a></li> <li>• <a href="#">lightlyActive</a></li> <li>• <a href="#">sedentaryActive</a></li> </ul> Endpoint(s): <a href="#">Get Daily Activity Summary</a>
duration	integer	The length in time (milliseconds) after the exercise was edited. If the exercise was not edited, the duration = originalDuration. This value will contain pauses during the exercise. Endpoint(s): <a href="#">Get Daily Activity Summary</a> , <a href="#">Get Frequent Activities</a> , <a href="#">Get Recent Activity Types</a>
elevation	integer	The elevation traveled for the day displayed in the units defined by the Accept-Language header. Endpoint(s): <a href="#">Get Daily Activity Summary</a>
elevationGain	integer	Elevation gained during the exercise. Endpoint(s): <a href="#">Get Activity Logs List</a>
fairlyActiveMinutes	integer	Total minutes the user was fairly/moderately active. Endpoint(s): <a href="#">Get Daily Activity Summary</a>
floors	integer	The equivalent floors climbed associated with the activity, goal, summary, and lifetime totals. Displayed



		<p>in the units defined by the Accept-Language header.</p> <p>Endpoint(s): <a href="#">Get Daily Activity Summary</a>, <a href="#">Get Activity Goals</a>, <a href="#">Get Lifetime Stats</a></p>
heartRateLink	string	<p>Link to fetch the intraday heart rate data for the activity recorded, using the date and time the activity took place.</p> <p>Endpoint(s): <a href="#">Get Activity Logs List</a></p>
heartRateZones	list	<p>Returns the minutes spent in, and min/max values, of each heart rate zone for:</p> <ul style="list-style-type: none"> <li>• <a href="#">Out of range</a></li> <li>• <a href="#">Fat Burn</a></li> <li>• <a href="#">Cardio</a></li> <li>• <a href="#">Peak</a></li> </ul> <p>Endpoint(s): <a href="#">Get Activity Logs List</a></p>
intensity	integer	<p>Integer representing the intensity level of the activity.</p> <ul style="list-style-type: none"> <li>• <a href="#">0 = Sedentary</a></li> <li>• <a href="#">1 = Lightly active</a></li> <li>• <a href="#">2 = Moderately/Fairly active</a></li> <li>• <a href="#">3 = Very active</a></li> </ul> <p>Endpoint(s): <a href="#">Get Activity TCX</a></p>
lap (starttime)	time	<p>Timestamp representing each exercise lap start time.</p> <p>Endpoint(s): <a href="#">Get Activity TCX</a></p>
latitudedegrees	integer	<p>The GPS latitude of the recorded exercise at the specified time</p> <p>Endpoint(s): <a href="#">Get Activity TCX</a></p>
longitudedegrees	integer	<p>The GPS longitude of the recorded exercise at the specified time.</p> <p>Endpoint(s): <a href="#">Get Activity TCX</a></p>
lightlyActiveMinutes	integer	<p>Total minutes the user was lightly active.</p> <p>Endpoint(s): <a href="#">Get Daily Activity Summary</a></p>



logType	string	<p>Method of which the activity was logged.</p> <p>Supported: manual   mobile_run   tracker   auto_detected   fitstar   or the &lt;name&gt; of the 3rd party application</p> <p>Endpoint(s): <a href="#">Get Activity Logs List</a></p>
manualValuesSpecified		<p>Manually logged counts for the following elements: calories, distance, steps.</p> <p>Endpoint(s): <a href="#">Get Activity Logs List</a></p>
marginalCalories	integer	<p>Estimated marginal calories burned.</p> <p>Endpoint(s): <a href="#">Get Daily Activity Summary</a></p>
mets	integer	<p>The metabolic equivalent (METs) of the activity performed.</p> <p>Endpoint(s): <a href="#">Get Activity Type</a>, <a href="#">Get Favorite Activities</a></p>
minutes	integer	<p>Total number of minutes the user spent during the specified activity level for that day.</p> <p>Endpoint(s): <a href="#">Get Activity Logs List</a></p>
name	string	<p>Name of the activity level.</p> <p>Supported: <a href="#">sedentary</a>   <a href="#">lightly</a>   <a href="#">fairly</a>   <a href="#">very</a></p> <p>Endpoint(s): <a href="#">Get Activity Logs List</a></p>
name	string	<p>Name of the recorded exercise.</p> <p>Endpoint(s): <a href="#">Get Daily Activity Summary</a></p>
originalDuration	integer	<p>The initial length in time (milliseconds) that the exercise was recorded. This value will contain pauses during the exercise.</p> <p>Endpoint(s): <a href="#">Get Activity Logs List</a></p>
originalStartTime	integer	<p>The initial start datetime that the exercise was recorded.</p> <p>Endpoint(s): <a href="#">Get Activity Logs List</a></p>



pace	integer	Calculated average pace during the exercise. Endpoint(s): <a href="#">Get Activity Logs List</a>
sedentaryMinutes	integer	Total minutes the user was <a href="#">sedentary</a> . Endpoint(s): <a href="#">Get Daily Activity Summary</a>
speed	integer	Average speed during the exercise. Endpoint(s): <a href="#">Get Activity Logs List</a>
startDate	date	The start date of the recorded exercise. Endpoint(s): <a href="#">Get Daily Activity Summary</a>
startTime	date/date Time	The start datetime after the exercise was edited. If the exercise was not edited, the startTime = originalStartTime. Some endpoints may also return the date and UTC offset. Endpoint(s): <a href="#">Get Daily Activity Summary</a> , <a href="#">Get Activity Logs List</a>
steps	integer	Step counts associated with the activity, goal, summary, and lifetime totals. Endpoint(s): <a href="#">Get Daily Activity Summary</a> , <a href="#">Get Activity Logs List</a> , <a href="#">Get Activity Goals</a> , <a href="#">Get Lifetime Stats</a>
time	time	The time that metrics were recorded during the exercise. Endpoint(s): <a href="#">Get Activity TCX</a>
totaltimeseconds	integer	Length of the exercise lap in seconds Endpoint(s): <a href="#">Get Activity TCX</a>
value	integer	The heart rate recorded at the specified timestamp. Endpoint(s): <a href="#">Get Activity TCX</a>
veryActiveMinutes	integer	Total minutes the user was very active.



		Endpoint(s): <a href="#">Get Daily Activity Summary</a>
--	--	---

## GET Activity Time Series

Element Name	Datatype	Description
activities-log-[resource]	string	<p>The resource can be one of the following values:</p> <p>Calories   Steps   Distance   Floors   Elevation   CaloriesBMR   minutesSedentary   minutesLightlyActive   minutesFairlyActive   minutesVeryActive   minutesActivityCalories.</p> <p>The selected resource will be returned with the summary value for the given date or date range.</p> <p>Endpoint(s): <a href="#">Get Activity Timeseries</a></p>
activities-[resource]	string	<p>The resource can be one of the following values:</p> <p>Calories   Steps   Distance   Floors   Elevation.</p> <p>The selected resource will be returned with the summary value for the given date and time range.</p> <p>Endpoint(s): <a href="#">Get Activity Intraday Timeseries</a></p>
dateTime	date	<p>Date of the requested resource; in the format <b>YYYY-MM-DD</b>.</p> <p>Endpoint(s): <a href="#">Get Activity Timeseries</a>, <a href="#">Get Activity Intraday Timeseries</a></p>
value	integer	<p>Total count of the requested resource.</p> <p>Endpoint(s): <a href="#">Get Activity Timeseries</a>, <a href="#">Get Activity Intraday Timeseries</a></p>





## GET Activity Intraday Time Series

Activity intraday data can be returned in intervals of 1 minute or 15 minutes for calories, steps, distance, floors and elevation. Intraday data is only available by [request](#) and approved on a case by case basis.

activities-[resource]	string	<p>Resource can be one of the following values:</p> <p>Calories   Steps   Distance   Floors   Elevation   activeZoneMinutes</p> <p>The selected resource will be returned with the summary value for the given date and time range.</p> <p>Endpoint(s): <a href="#">Get Activity Intraday Timeseries</a></p>
dateTime	date	<p>Date of the requested resource; in the format <b>YYYY-MM-DD</b>.</p> <p>Endpoint(s): <a href="#">Get Activity Intraday Timeseries</a></p>
level	integer	<p>Numerical value representing the user's activity-level at the moment when the resource was recorded.</p> <p><a href="#">0 = sedentary</a>  <a href="#">1 = lightly active</a>  <a href="#">2 = fairly/moderately active</a>  <a href="#">3 = very active</a></p> <p>Returned only when resource = calories</p> <p>Endpoint(s): <a href="#">Get Activity Intraday Timeseries</a></p>
mets	integer	<p>METs value at the moment when the resource was recorded.</p> <p>Returned only when resource = calories</p> <p>Endpoint(s): <a href="#">Get Activity Intraday Timeseries</a></p>
time	time	<p>The time the resource was recorded; in the format <b>HH:MM</b></p> <p>Endpoint(s): <a href="#">Get Activity Intraday Timeseries</a></p>



value	integer	The specified resource's value at the time it is recorded  Endpoint(s): <a href="#">Get Activity Intraday Timeseries</a>
-------	---------	--

## Active Zone Minutes (AZM)

The Active Zone Minutes (AZM) endpoints are used for querying the user's heart-pumping activity throughout the day.

### GET AZM

Element Name	Datatype	Description
dateTime	date	Date of log; in the format <b>YYYY-MM-DD</b> or <b>today</b> .  Endpoint(s): <a href="#">Get AZM Daily Summary by Date</a> , <a href="#">Get AZM Daily Summary by Interval</a>
activeZoneMinutes	integer	Total count of active zone minutes  Endpoint(s): <a href="#">Get AZM Daily Summary by Date</a> , <a href="#">Get AZM Daily Summary by Interval</a>
fatBurnActiveZoneMinutes	integer	The number of active zone minutes in the fat burn heart rate zone. 1 fat burn minute = 1 fat burn active zone minute.  Endpoint(s): <a href="#">Get AZM Daily Summary by Date</a> , <a href="#">Get AZM Daily Summary by Interval</a>
cardioActiveZoneMinutes	integer	The number of active zone minutes in the cardio heart rate zone. 1 cardio minute = 2 cardio active zone minutes.  Endpoint(s): <a href="#">Get AZM Daily Summary by Date</a> , <a href="#">Get AZM Daily Summary by Interval</a>
peakActiveZoneMinutes	integer	The number of active zone minutes in the peak heart rate zone. 1 peak minute = 2 peak active zone minutes.



		Endpoint(s): <a href="#">Get AZM Daily Summary by Date</a> , <a href="#">Get AZM Daily Summary by Interval</a>
--	--	--

## GET AZM Intraday Time Series

Element Name	Datatype	Description
dateTime	date	Date of log; in the format <b>YYYY-MM-DD</b> or <b>today</b> .  Endpoint(s): <a href="#">Get AZM Intraday by Date</a> , <a href="#">Get AZM Intraday by Interval</a>
minute	dateTime	The dateTime when the AZM value was recorded.  Endpoint(s): <a href="#">Get AZM Intraday by Date</a> , <a href="#">Get AZM Intraday by Interval</a>
fatBurnActiveZoneMinutes	integer	The number of active zone minutes in the fat burn heart rate zone earned during the previous iteration. 1 fat burn minute = 1 fat burn active zone minute.  Endpoint(s): <a href="#">Get AZM Intraday by Date</a> , <a href="#">Get AZM Intraday by Interval</a>
cardioActiveZoneMinutes	integer	The number of active zone minutes in the cardio heart rate zone earned during the previous iteration. 1 cardio minute = 2 cardio active zone minutes.  Endpoint(s): <a href="#">Get AZM Intraday by Date</a> , <a href="#">Get AZM Intraday by Interval</a>
peakActiveZoneMinutes	integer	The number of active zone minutes in the peak heart rate zone earned during the previous iteration. 1 peak minute = 2 peak active zone minutes.  Endpoint(s): <a href="#">Get AZM Intraday by Date</a> , <a href="#">Get AZM Intraday by Interval</a>
activeZoneMinutes	integer	Total count of active zone minutes earned during the previous iteration.  Endpoint(s): <a href="#">Get AZM Intraday by Date</a> , <a href="#">Get AZM Intraday by Interval</a>



## Body & Weight

### GET Body & Weight

Element Name	Datatype	Description
bmi	float	Calculated BMI in the format <b>X.XX</b> . Body mass index is a value derived from the mass and height of a person.  Endpoint(s): <a href="#">Get Weight Logs</a>
date	date	The date in the format <b>yyyy-MM-dd</b> .  Endpoint(s): <a href="#">Get Body Fat Logs</a> , <a href="#">Get Weight Logs</a>
fat	float	Body fat percentage; in the format <b>X.XX</b> . <ul style="list-style-type: none"><li>Returned only when goal type is set to <b>fat</b>.*</li></ul> Endpoint(s): <a href="#">Get Body Fat Logs</a> , <a href="#">Get Body Goals</a> *, <a href="#">Get Weight Logs</a>
source	string	The source of the weight log.  Supported: API   Aria   AriaAir   Withings  Endpoint(s): <a href="#">Get Body Fat Logs</a> , <a href="#">Get Weight Logs</a>
startDate	date	The start date of the body goal; in the format <b>YYYY-MM-DD</b> . <ul style="list-style-type: none"><li>Returned only when goal type is set to <b>weight</b>.</li></ul> Endpoint(s): <a href="#">Get Body Goals</a>
startWeight	integer	The user's recorded weight on the goal startDate in the unit system that corresponds to the Accept-Language header provided or if not provided in metric. <ul style="list-style-type: none"><li>Returned only when goal type is set to <b>weight</b>.</li></ul>



		Endpoint(s): <a href="#">Get Body Goals</a>
time	time	Time of the measurement; hours and minutes in the format <b>HH:mm:ss</b> , set to the last second of the day if not provided.  Endpoint(s): <a href="#">Get Body Fat Logs</a> , <a href="#">Get Weight Logs</a>
weight	float	Weight in the format <b>X.XX</b> , in the unit system that corresponds to the Accept-Language header provided or if not provided in metric.  <ul style="list-style-type: none"> <li>Returned only when goal type is set to <b>weight</b>.*</li> </ul> Endpoint(s): <a href="#">Get Body Goals</a> *, <a href="#">Get Weight Logs</a>

## GET Body Time Series

The **Get Body Time Series API** returns time series data in the specified range for a given resource in the format requested using units in the unit systems that corresponds to the Accept-Language header provided.

Element Name	Datatype	Description
body-[resource]	string	The resource can be one of the following values:  BMI   FAT   WEIGHT  The selected resource will be returned with the summary value for the given date and time range.  Endpoint(s): <a href="#">Get Body Time Series</a>
dateTime	date	Date of the requested resource; in the format <b>YYYY-MM-DD</b> .  Endpoint(s): <a href="#">Get Body Time Series</a>
value	float	The value recorded at the specific timestamp.  Endpoint(s): <a href="#">Get Body Time Series</a>



## Breathing Rate

Breathing rate data applies specifically to a user's "main sleep," which is the longest period of time asleep on a given date.

### GET Breathing Rate

Element Name	Datatype	Description
br : dateTime	date	Date of log; in the format <b>YYYY-MM-DD</b> .  Endpoint(s): <a href="#">Get Breathing Rate Summary by Date</a> , <a href="#">Get Breathing Rate Summary by Interval</a>
br : value : breathingRate	float	Average number of breaths taken per minute.  Endpoint(s): <a href="#">Get Breathing Rate Summary by Date</a> , <a href="#">Get Breathing Rate Summary by Interval</a>

### GET Breathing Rate Intraday Time Series

The data returned includes intraday data for a specified date or date range. It measures your average breathing rate throughout the day and categories your breathing rate by sleep stage.

Intraday data is only available by [request](#) and approved on a case by case basis.

Element Name	Datatype	Description
br : dateTime	date	Date of log; in the format <b>YYYY-MM-DD</b> .  Endpoint(s): <a href="#">Get Breathing Rate Intraday by Date</a> , <a href="#">Get Breathing Rate Intraday by Interval</a>
br : value : lightSleepSummary : breathingRate	float	Average number of breaths taken per minute when the user was in light sleep.  Endpoint(s): <a href="#">Get Breathing Rate Intraday by Date</a> , <a href="#">Get Breathing Rate Intraday by Interval</a>
br : value : deepSleepSummary : breathingRate	float	Average number of breaths taken per minute when the user was in deep sleep.



		Endpoint(s): <a href="#">Get Breathing Rate Intraday by Date</a> , <a href="#">Get Breathing Rate Intraday by Interval</a>
br : value : remSleepSummary : breathingRate	integer	Average number of breaths taken per minute when the user was in REM sleep.  Endpoint(s): <a href="#">Get Breathing Rate Intraday by Date</a> , <a href="#">Get Breathing Rate Intraday by Interval</a>
br : value : fullSleepSummary : breathingRate	float	Average number of breaths taken per minute throughout the entire period of sleep which you can compare to the sleep stage-specific measurements.  Endpoint(s): <a href="#">Get Breathing Rate Intraday by Date</a> , <a href="#">Get Breathing Rate Intraday by Interval</a>

## Cardio Fitness Score (VO2 Max)

### GET VO2 Max Summary

cardioscore : dateTime	date	The date specified in the format YYYY-MM-DD.  Endpoint(s): <a href="#">Get VO2 Max Summary by Date</a> , <a href="#">Get VO2 Max Summary by Interval</a>
cardioscore : value : vo2Max	numeric (range)	The displayable value of VO2 Max in mL/kg/min.  Endpoint(s): <a href="#">Get VO2 Max Summary by Date</a> , <a href="#">Get VO2 Max Summary by Interval</a>

## Devices

### GET Devices

battery	integer	Numerical value representing the percentage of the device's battery life.
---------	---------	---



		Supported: 0-100 Endpoint(s): <a href="#">Get Devices</a>
deviceVersion	string	Name of the device  Supported: Ace   Ace 2   Alta   Aria   Aria Air   Aria 2   Blaze   Charge   Charge 2   Charge 3   Charge 4   Charge HR   Classic   Flex   Flex 2   Inspire   Inspire 2   Inspire HR   Ionic   One   Sense   Ultra   Versa   Versa 2   Versa 3   Zip  Endpoint(s): <a href="#">Get Devices</a>
enabled	boolean	<b>true</b> or <b>false</b> ; if false, alarm does not vibrate until enabled is set to true  Supported: True   False  Endpoint(s): <a href="#">Get Alarms</a>
features	string	Placeholder for displaying the device features. At this time, an empty string is returned.  Endpoint(s): <a href="#">Get Devices</a>
lastSyncTime	date	The time the device last synced with the Fitbit mobile app; returned in the format <b>YYYY-DD-MMTHH:MM:SS:FFF</b>  Endpoint(s): <a href="#">Get Devices</a>
recurring	boolean	<b>true</b> or <b>false</b> ; if false, the alarm is a single event.  Supported: True   False  Endpoint(s): <a href="#">Get Alarms</a>
snoozeCount	integer	Maximum number of times a user can snooze the alarm.  Endpoint(s): <a href="#">Get Alarms</a>
snoozeLength	integer	Number of minutes until the next snooze alarm occurs.  Endpoint(s): <a href="#">Get Alarms</a>





syncedToDevice	boolean	States if the alarm is synced to the tracker after it was created in the Fitbit mobile app.  Supported: True   False  Endpoint(s): <a href="#">Get Alarms</a>
time	time	Time of day that the alarm vibrates with a UTC timezone offset, e.g. 07:15-08:00  Endpoint(s): <a href="#">Get Alarms</a>
type	string	Type of device  Supported: Tracker   Scale  Endpoint(s): <a href="#">Get Devices</a>
vibe	string	Vibration pattern  Supported: DEFAULT  Endpoint(s): <a href="#">Get Alarms</a>
weekDays	list	List of days of the week on which the alarm vibrates  Supported: Sunday   Monday   Tuesday   Wednesday   Thursday   Friday   Saturday  Endpoint(s): <a href="#">Get Alarms</a>

## Electrocardiogram (ECG)

The Electrocardiogram (also known as ECG) endpoint is used for querying the user's on-device ECG readings.

### GET ECG

This endpoint retrieves a list of the user's Electrocardiogram (ECG) log entries before or after a given day.

Element Name	Datatype	Description
--------------	----------	-------------



afterDate	dateTime	The afterDate parameter of the request. Endpoint(s): <a href="#">Get ECG Log List</a>
averageHeartRate	integer	The average heart rate of the user during the recording. Endpoint(s): <a href="#">Get ECG Log List</a>
beforeDate	dateTime	The beforeDate parameter of the request. Endpoint(s): <a href="#">Get ECG Log List</a>
deviceName	string	Hardware name of the device used to take the measurement. Endpoint(s): <a href="#">Get ECG Log List</a>
leadNumber	integer	The ECG lead being used to take the reading. For the Fitbit devices, the leadNumber will always be 1. <b>Supported:</b> 1 Endpoint(s): <a href="#">Get ECG Log List</a>
numberOfWaveformSamples	integer	The total number of samples in the recording. Endpoint(s): <a href="#">Get ECG Log List</a>
previous	string	The URL of the request that will fetch the previous page of results. Endpoint(s): <a href="#">Get ECG Log List</a>
resultClassification	string	See Classification of ECG results for more information. <b>Supported:</b> <a href="#">Atrial Fibrillation</a>   <a href="#">Normal Sinus Rhythm</a>   <a href="#">Inconclusive</a>   <a href="#">Inconclusive: High heart rate</a>   <a href="#">Inconclusive: Low heart rate</a> Endpoint(s): <a href="#">Get ECG Log List</a>
startTime	dateTime	The date and time when the reading was started on the device.



		Endpoint(s): <a href="#">Get ECG Log List</a>
samplingFrequencyHz	integer	The frequency in hertz at which Fitbit sampled the voltage.  <b>Supported:</b> 250  Endpoint(s): <a href="#">Get ECG Log List</a>
scalingFactor	integer	The scaling factor used to convert waveform samples to ECG voltages in mV (mV = sample / scalingFactor).  <b>Supported:</b> 10922  Endpoint(s): <a href="#">Get ECG Log List</a>
waveFormSamples	array	An array of integers representing the ECG waveform visible to the user on their PDF report.  Endpoint(s): <a href="#">Get ECG Log List</a>

## Food Logging

### GET Food & Water

amount	integer	Quantity of the food or water entry displayed in the units defined by the <a href="#">Accept-Language</a> header.  Endpoint(s): <a href="#">Get Food Logs</a> , <a href="#">Get Water Logs</a> , <a href="#">Get Frequent Foods</a> , <a href="#">Get Recent Foods</a>
barcode	boolean	Value determining if barcode was provided.  Supported: TRUE   FALSE  Endpoint(s): <a href="#">Get Food Locales</a>
brand	string	Brand of the food logged  Endpoint(s): <a href="#">Get Food Logs</a> , <a href="#">Get Favorite Foods</a> , <a href="#">Get Frequent Foods</a> , <a href="#">Get Recent Foods</a> , <a href="#">Get Food</a>
carbs	float	Amount of carbohydrates in the recorded food entry;



		<p>returned in <b>grams</b>. For a complete list...</p> <p>Endpoint(s): <a href="#">Get Food Logs</a>, <a href="#">Get Favorite Foods</a>, <a href="#">Get Food</a></p>
calories	integer	<p>Value for daily calorie consumption.</p> <p>Endpoint(s): <a href="#">Get Food Goals</a>, <a href="#">Get Food Logs</a>, <a href="#">Get Favorite Foods</a>, <a href="#">Get Frequent Foods</a>, <a href="#">Get Recent Foods</a>, <a href="#">Get Food</a>, <a href="#">Get Search Foods</a></p>
dateLastEaten	date	<p>Date of when the food was last consumed.</p> <p>Endpoint(s): <a href="#">Get Frequent Foods</a></p>
defaultServingSize	integer	<p>The default serving size associated with the food log.</p> <p>Endpoint(s): <a href="#">Get Favorite Foods</a>, <a href="#">Get Food</a>, <a href="#">Get Search Foods</a></p>
defaultUnit	list	<p>Returns information associated to the unit specified:</p> <ul style="list-style-type: none"> <li>• <b>id</b>: numerical identifier for the unit</li> <li>• <b>name</b>: name of the unit</li> <li>• <b>plural</b>: abbreviation of the unit</li> </ul> <p>Endpoint(s): <a href="#">Get Favorite Foods</a>, <a href="#">Get Food</a>, <a href="#">Get Search Foods</a></p>
estimatedCaloriesOut	integer	<p>Number of calories expected to burn from <a href="#">BMR</a> plus the calories needed to burn through activity based on the <a href="#">food plan intensity</a> goal.</p> <p>Example: If BMR calories = 800 and Food Plan intensity is "Harder" (1000 calories), estimated calories out is 800 + 1000 = 1800</p> <p>Endpoint(s): <a href="#">Get Food Logs</a></p>
estimatedDate	date	<p>Estimated date of goal completion; in the format <b>yyyy-MM-dd</b>.</p> <ul style="list-style-type: none"> <li>• Returned only when intensity type is set to one of the following:</li> </ul> <p><a href="#">EASIER</a>   <a href="#">MEDIUM</a>   <a href="#">KINDA HARD</a>   <a href="#">HARDER</a></p>



		Endpoint(s): <a href="#">Get Food Goals</a>
fat	float	Amount of fat in the recorded food entry; returned in <b>grams</b> .  Endpoint(s): <a href="#">Get Food Logs</a> , <a href="#">Get Favorite Foods</a> , <a href="#">Get Food</a>
fiber	float	Amount of fiber in the recorded food entry; returned in <b>grams</b> .  Endpoint(s): <a href="#">Get Food Logs</a> , <a href="#">Get Favorite Foods</a> , <a href="#">Get Food</a>
goal	integer	Amount of water to be consumed set by the user's goal; returned in milliliters.  Endpoint(s): <a href="#">Get Water Goal</a>
intensity	string	Level of difficulty for the selected food plan.  Supported: <a href="#">MAINTENANCE</a>   <a href="#">EASIER</a>   <a href="#">MEDIUM</a>   <a href="#">KINDA HARD</a>   <a href="#">HARDER</a>  Endpoint(s): <a href="#">Get Food Goals</a>
isFavorite	boolean	Value determining if the selected food is categorized as one of the user's favorite foods.  Supported: TRUE   FALSE  Endpoint(s): <a href="#">Get Food Logs</a>
mealTypeld	integer	Numerical value associated with when the food was consumed.  <ul style="list-style-type: none"> <li>● 1=Breakfast</li> <li>● 2=Morning Snack</li> <li>● 3=Lunch</li> <li>● 4=Afternoon Snack</li> <li>● 5=Dinner</li> <li>● 7=Anytime.</li> </ul> Endpoint(s): <a href="#">Get Frequent Foods</a>
name	string	Name of the food.



		Endpoint(s): <a href="#">Get Food Logs</a> , <a href="#">Get Frequent Foods</a> , <a href="#">Get Recent Foods</a> , <a href="#">Get Food</a> , <a href="#">Get Search Foods</a> , <a href="#">Get Favorite Foods</a>
personalized	boolean	Specifies if the user personalized the selected food plan.  Supported: TRUE   FALSE  Endpoint(s): <a href="#">Get Food Goals</a>
protein	float	Amount of protein in the recorded food entry; returned in <b>grams</b> .  Endpoint(s): <a href="#">Get Food Logs</a> , <a href="#">Get Favorite Foods</a> , <a href="#">Get Food</a>
servings	list	Metadata related to the food's servings:  <ul style="list-style-type: none"> <li>• <b>Multiplier</b>: Number of times the serving size was consumed.</li> <li>• <b>servinSize</b>: The number of servings within the food product.</li> </ul> Endpoint(s): <a href="#">Get Favorite Foods</a> , <a href="#">Get Food</a>
sodium	float	Amount of Sodium in the recorded food entry; returned in <b>grams</b> .  Endpoint(s): <a href="#">Get Food Logs</a> , <a href="#">Get Favorite Foods</a> , <a href="#">Get Food</a>
startDate	date	Start date of the water goal entry.  Endpoint(s): <a href="#">Get Water Goal</a>
water	float	Daily water consumption displayed in the units defined by the <a href="#">Accept-Language</a> header.  Endpoint(s): <a href="#">Get Water Logs</a> , <a href="#">Get Food Logs</a>

## GET Food & Water Time Series



dateTime	date	Date of the requested resource; in the format <b>YYYY-MM-DD</b> . Endpoint(s): <a href="#">Get Food or Water Timeseries</a>
resource-path	string	The resource can be one of the following values:  Supported: caloriesIn   water  The selected resource will be returned with the summary value for the given date or date range.  Endpoint(s): <a href="#">Get Food or Water Timeseries</a>
value	integer	Total count of the requested resource.  Endpoint(s): <a href="#">Get Food or Water Timeseries</a>

## Nutritional Values

biotin	integer	Amount of biotin in the recorded food entry; returned in <b>grams</b> .  Endpoint(s): <a href="#">Get Food Logs</a> , <a href="#">Get Favorite Foods</a> , <a href="#">Get Food</a>
calcium	integer	Amount of calcium in the recorded food entry; returned in <b>grams</b> .  Endpoint(s): <a href="#">Get Food Logs</a> , <a href="#">Get Favorite Foods</a> , <a href="#">Get Food</a>
calories	integer	Value for daily calorie consumption.  Endpoint(s): <a href="#">Get Food Goals</a> , <a href="#">Get Food Logs</a> , <a href="#">Get Favorite Foods</a> , <a href="#">Get Frequent Foods</a> , <a href="#">Get Recent Foods</a> , <a href="#">Get Food</a> , <a href="#">Get Search Foods</a>
caloriesFromFat	integer	Amount of calories from fat in the recorded food entry; returned in <b>grams</b> .  Endpoint(s): <a href="#">Get Food Logs</a> , <a href="#">Get Favorite Foods</a> , <a href="#">Get Food</a>



cholesterol	integer	Amount of cholesterol in the recorded food entry; returned in <b>grams</b> . Endpoint(s): <a href="#">Get Food Logs</a> , <a href="#">Get Favorite Foods</a> , <a href="#">Get Food</a>
copper	integer	Amount of copper in the recorded food entry; returned in <b>grams</b> . Endpoint(s): <a href="#">Get Food Logs</a> , <a href="#">Get Favorite Foods</a> , <a href="#">Get Food</a>
dietaryFiber	integer	Amount of dietaryFiber in the recorded food entry; returned in <b>grams</b> . Endpoint(s): <a href="#">Get Food Logs</a> , <a href="#">Get Favorite Foods</a> , <a href="#">Get Food</a>
fat	float	Amount of fat in the recorded food entry; returned in <b>grams</b> . Endpoint(s): <a href="#">Get Food Logs</a> , <a href="#">Get Favorite Foods</a> , <a href="#">Get Food</a>
fiber	float	Amount of fiber in the recorded food entry; returned in <b>grams</b> . Endpoint(s): <a href="#">Get Food Logs</a> , <a href="#">Get Favorite Foods</a> , <a href="#">Get Food</a>
folicAcid	integer	Amount of folicAcid in the recorded food entry; returned in <b>grams</b> . Endpoint(s): <a href="#">Get Food Logs</a> , <a href="#">Get Favorite Foods</a> , <a href="#">Get Food</a>
iodine	integer	Amount of iodine in the recorded food entry; returned in <b>grams</b> . Endpoint(s): <a href="#">Get Food Logs</a> , <a href="#">Get Favorite Foods</a> , <a href="#">Get Food</a>
iron	integer	Amount of iron in the recorded food entry; returned in <b>grams</b> . Endpoint(s): <a href="#">Get Food Logs</a> , <a href="#">Get Favorite Foods</a> , <a href="#">Get Food</a>





magnesium	integer	Amount of magnesium in the recorded food entry; returned in <b>grams</b> .  Endpoint(s): <a href="#">Get Food Logs</a> , <a href="#">Get Favorite Foods</a> , <a href="#">Get Food</a>
niacin	integer	Amount of niacin in the recorded food entry; returned in <b>grams</b> .  Endpoint(s): <a href="#">Get Food Logs</a> , <a href="#">Get Favorite Foods</a> , <a href="#">Get Food</a>
pantothenicAcid	integer	Amount of pantothenicAcid in the recorded food entry; returned in <b>grams</b> .  Endpoint(s): <a href="#">Get Food Logs</a> , <a href="#">Get Favorite Foods</a> , <a href="#">Get Food</a>
phosphorus	integer	Amount of phosphorus in the recorded food entry; returned in <b>grams</b> .  Endpoint(s): <a href="#">Get Food Logs</a> , <a href="#">Get Favorite Foods</a> , <a href="#">Get Food</a>
potassium	integer	Amount of potassium in the recorded food entry; returned in <b>grams</b> .  Endpoint(s): <a href="#">Get Food Logs</a> , <a href="#">Get Favorite Foods</a> , <a href="#">Get Food</a>
protein	float	Amount of protein in the recorded food entry; returned in <b>grams</b> .  Endpoint(s): <a href="#">Get Food Logs</a> , <a href="#">Get Favorite Foods</a> , <a href="#">Get Food</a>
riboflavin	integer	Amount of riboflavin in the recorded food entry; returned in <b>grams</b> .  Endpoint(s): <a href="#">Get Food Logs</a> , <a href="#">Get Favorite Foods</a> , <a href="#">Get Food</a>
saturatedFat	integer	Amount of saturated fat in the recorded food entry; returned in <b>grams</b> .  Endpoint(s): <a href="#">Get Food Logs</a> , <a href="#">Get Favorite Foods</a> , <a href="#">Get Food</a>



sodium	float	Amount of Sodium in the recorded food entry; returned in <b>grams</b> .  Endpoint(s): <a href="#">Get Food Logs</a> , <a href="#">Get Favorite Foods</a> , <a href="#">Get Food</a>
sugars	integer	Amount of sugars in the recorded food entry; returned in <b>grams</b> .  Endpoint(s): <a href="#">Get Food Logs</a> , <a href="#">Get Favorite Foods</a> , <a href="#">Get Food</a>
thiamin	integer	Amount of thiamin in the recorded food entry; returned in <b>grams</b> .  Endpoint(s): <a href="#">Get Food Logs</a> , <a href="#">Get Favorite Foods</a> , <a href="#">Get Food</a>
totalCarbohydrate	integer	Amount of total carbohydrates in the recorded food entry; returned in <b>grams</b> .  Endpoint(s): <a href="#">Get Food Logs</a> , <a href="#">Get Favorite Foods</a> , <a href="#">Get Food</a>
totalFat	integer	Amount of total fat in the recorded food entry; returned in <b>grams</b> .  Endpoint(s): <a href="#">Get Food Logs</a> , <a href="#">Get Favorite Foods</a> , <a href="#">Get Food</a>
transFat	integer	Amount of trans fat in the recorded food entry; returned in <b>grams</b> .  Endpoint(s): <a href="#">Get Food Logs</a> , <a href="#">Get Favorite Foods</a> , <a href="#">Get Food</a>
vitaminA	integer	Amount of vitamin A in the recorded food entry; returned in <b>grams</b> .  Endpoint(s): <a href="#">Get Food Logs</a> , <a href="#">Get Favorite Foods</a> , <a href="#">Get Food</a>
vitaminB12	integer	Amount of vitamin B12 in the recorded food entry; returned in <b>grams</b> .  Endpoint(s): <a href="#">Get Food Logs</a> , <a href="#">Get Favorite Foods</a> , <a href="#">Get Food</a>



vitaminB6	integer	Amount of vitamin B6 in the recorded food entry; returned in <b>grams</b> .  Endpoint(s): <a href="#">Get Food Logs</a> , <a href="#">Get Favorite Foods</a> , <a href="#">Get Food</a>
vitaminC	integer	Amount of vitamin C in the recorded food entry; returned in <b>grams</b> .  Endpoint(s): <a href="#">Get Food Logs</a> , <a href="#">Get Favorite Foods</a> , <a href="#">Get Food</a>
vitaminD	integer	Amount of vitamin D in the recorded food entry; returned in <b>grams</b> .  Endpoint(s): <a href="#">Get Food Logs</a> , <a href="#">Get Favorite Foods</a> , <a href="#">Get Food</a>
vitaminE	integer	Amount of vitamin E in the recorded food entry; returned in <b>grams</b> .  Endpoint(s): <a href="#">Get Food Logs</a> , <a href="#">Get Favorite Foods</a> , <a href="#">Get Food</a>
zinc	integer	Amount of zinc in the recorded food entry; returned in <b>grams</b> .  Endpoint(s): <a href="#">Get Food Logs</a> , <a href="#">Get Favorite Foods</a> , <a href="#">Get Food</a>

## Heart

### GET Heart Rate Time Series

caloriesOut	integer	Number of calories burned with <a href="#">custom</a> and non-custom <a href="#">heart rate zones</a> .  Endpoint(s): <a href="#">Get Heart Rate Time Series</a>
dateTime	date	Date of the heart rate log; returned in the format <b>YYYY-DD-MM</b> .



		Endpoint(s): <a href="#">Get Heart Rate Time Series</a>
max	integer	Maximum heart rate range value for <a href="#">custom</a> and non-custom <a href="#">heart rate zones</a> . Endpoint(s): <a href="#">Get Heart Rate Time Series</a>
min	integer	Minimum heart rate range value for <a href="#">custom</a> and non-custom <a href="#">heart rate zones</a> . Endpoint(s): <a href="#">Get Heart Rate Time Series</a>
minutes	integer	Number of minutes within <a href="#">custom</a> and non-custom <a href="#">heart rate zones</a> . Endpoint(s): <a href="#">Get Heart Rate Time Series</a>
name	string	Name of the <a href="#">custom</a> and non-custom <a href="#">heart rate zones</a> . Endpoint(s): <a href="#">Get Heart Rate Time Series</a>
restingHeartRate	integer	<a href="#">BPM</a> value while at complete rest. RHR uses heart-rate data from both awake and asleep states to estimate RHR. Endpoint(s): <a href="#">Get Heart Rate Time Series</a>

## GET Heart Rate Intraday Time Series

Heart rate intraday data can be returned in intervals of 1 second or 1 minute. Intraday data is only available by [request](#) and approved on a case by case basis.

caloriesOut	integer	Number of calories burned with <a href="#">custom</a> and non-custom <a href="#">heart rate zones</a> . Endpoint(s): <a href="#">Get Heart Rate Intraday Time Series</a>
dateTime	date	Date of the heart rate log; returned in the format <b>YYYY-MM-DD</b> .



		Endpoint(s): <a href="#">Get Heart Rate Intraday Time Series</a>
max	integer	Maximum heart rate range value for <a href="#">custom</a> and non-custom <a href="#">heart rate zones</a> .  Endpoint(s): <a href="#">Get Heart Rate Time Series</a> , <a href="#">Get Heart Rate Intraday Time Series</a>
min	integer	Minimum heart rate range value for <a href="#">custom</a> and non-custom <a href="#">heart rate zones</a> .  Endpoint(s): <a href="#">Get Heart Rate Intraday Time Series</a>
minutes	integer	Number of minutes within the <a href="#">custom</a> and non-custom <a href="#">heart rate zones</a> .  Endpoint(s): <a href="#">Get Heart Rate Intraday Time Series</a>
name	string	Name of the <a href="#">custom</a> and non-custom <a href="#">heart rate zones</a> .  Endpoint(s): <a href="#">Get Heart Rate Intraday Time Series</a>
restingHeartRate	integer	Resting heart rate value for the day. RHR uses heart-rate data from both awake and asleep states to estimate RHR.  Endpoint(s): <a href="#">Get Heart Rate Intraday Time Series</a>
time	time	The time the intraday heart rate value was recorded; returned in the format <b>HH:MM:SS</b> .  Endpoint(s): <a href="#">Get Heart Rate Intraday Time Series</a>
value	integer	This element will represent one of the following: <ul style="list-style-type: none"> <li>1. The heart rate value at the time the reading was recorded.</li> <li>2. The average sum of the heart rate values in the activities-heart-intraday dataset. It will <u>return only when start-time and end-time are specified in the endpoint.</u></li> </ul>



		Endpoint(s): <a href="#">Get Heart Rate Intraday Time Series</a>
--	--	--

## HRV

The following information is returned from the Heart Rate Variability (HRV) endpoints, logged by Fitbit devices worn by Fitbit users. HRV data applies specifically to a user’s “main sleep,” which is the longest single period of time asleep on a given date.

### GET HRV

Element Name	Datatype	Description
dateTime	date	Date of log; in the format <b>YYYY-MM-DD</b> .  Endpoint(s): <a href="#">Get HRV Summary by Date</a> , <a href="#">Get HRV Summary by Interval</a>
value : dailyRmssd	float	The Root Mean Square of Successive Differences ( <b>RMSSD</b> ) between heart beats. It measures short-term variability in the user’s daily heart rate in milliseconds (ms).  Endpoint(s): <a href="#">Get HRV Summary by Date</a> , <a href="#">Get HRV Summary by Interval</a>
value : deepRmssd	float	The Root Mean Square of Successive Differences ( <b>RMSSD</b> ) between heart beats. It measures short-term variability in the user’s heart rate while in deep sleep, in milliseconds (ms).  Endpoint(s): <a href="#">Get HRV Summary by Date</a> , <a href="#">Get HRV Summary by Interval</a>

### GET HRV Intraday Time Series

This information returns the Heart Rate Variability (HRV) intraday data for a single date or date range. HRV data applies specifically to a user’s “main sleep,” which is the longest single period of time asleep on a given date.

Intraday data is only available by [request](#) and approved on a case by case basis.



Element Name	Datatype	Description
hrv : dateTime	date	Date of log; in the format <b>YYYY-MM-DD</b> .  Endpoint(s): <a href="#">Get HRV Intraday by Date</a> , <a href="#">Get HRV Intraday by Interval</a>
hrv : minutes : minute	dateTime	A measurement taken at a given time.  Endpoint(s): <a href="#">Get HRV Intraday by Date</a> , <a href="#">Get HRV Intraday by Interval</a>
hrv : minutes : value : rmssd	float	The Root Mean Square of Successive Differences ( <b>RMSSD</b> ) between heart beats. It measures short-term variability in the user's heart rate in milliseconds (ms).  Endpoint(s): <a href="#">Get HRV Intraday by Date</a> , <a href="#">Get HRV Intraday by Interval</a>
hrv : minutes : value : coverage	float	Data completeness in terms of the number of interbeat intervals.  Endpoint(s): <a href="#">Get HRV Intraday by Date</a> , <a href="#">Get HRV Intraday by Interval</a>
hrv : minutes : value : hf	float	The power in interbeat interval fluctuations within the high frequency band (0.15 Hz - 0.4 Hz).  Endpoint(s): <a href="#">Get HRV Intraday by Date</a> , <a href="#">Get HRV Intraday by Interval</a>
hrv : minutes : value : lf	float	The power in interbeat interval fluctuations within the high frequency band (0.04 Hz - 0.15 Hz).  Endpoint(s): <a href="#">Get HRV Intraday by Date</a> , <a href="#">Get HRV Intraday by Interval</a>

## Sleep

### GET Sleep Logs



count	integer	Total count of how many times a user was in the associated sleep stage.  Endpoint(s): <a href="#">Get Sleep Logs</a> , <a href="#">Get Sleep Logs by Date Range</a> , <a href="#">Get Sleep Logs List</a>
dateOfSleep	date	Date of recorded sleep; returned in <b>YYYY-DD-MM</b> .  Endpoint(s): <a href="#">Get Sleep Logs</a> , <a href="#">Get Sleep Logs by Date Range</a> , <a href="#">Get Sleep Logs List</a>
dateTime	dateTime	Timestamp when the user enters the sleep stage  Endpoint(s): <a href="#">Get Sleep Logs</a> , <a href="#">Get Sleep Logs by Date Range</a> , <a href="#">Get Sleep Logs List</a>
duration	integer	Duration of sleep log in milliseconds.  Endpoint(s): <a href="#">Get Sleep Logs</a> , <a href="#">Get Sleep Logs by Date Range</a> , <a href="#">Get Sleep Logs List</a>
efficiency	integer	An algorithm based on collected sleep metrics to determine how efficient the user slept.  Endpoint(s): <a href="#">Get Sleep Logs</a> , <a href="#">Get Sleep Logs by Date Range</a> , <a href="#">Get Sleep Logs List</a>
endTime	dateTime	Timestamp when the sleep log ended; returned in <b>YYYY-DD-MMTHH:MM:SS:FFF</b> .  Endpoint(s): <a href="#">Get Sleep Logs</a>
flowId	integer	An integer value representing the sleep goal consistency flow.  <ul style="list-style-type: none"><li>● <b>0</b> = A sleep goal is set, but there are not enough sleep logs recorded.</li><li>● <b>1</b> = The user either missed their sleep goal or no goal is set, but there are enough sleep logs recorded.</li><li>● <b>2</b> = A sleep goal is not set, and there are not enough sleep logs recorded.</li><li>● <b>3</b> = The user achieved their sleep goal.</li></ul>





		Endpoint(s): <a href="#">Get Sleep Goals</a>
infoCode	integer	<p>An integer value representing the quality of data collected within the sleep log.</p> <ul style="list-style-type: none"> <li>• <b>0</b> = Sufficient data to generate a sleep log.</li> <li>• <b>1</b> = Insufficient heart rate data.</li> <li>• <b>2</b> = Sleep period was too short (less than 3 hours).</li> <li>• <b>3</b> = Server-side issue</li> </ul> <p>Endpoint(s): <a href="#">Get Sleep Logs List</a></p>
isMainSleep	boolean	<p>States if the sleep record was the longest sleep of the day.</p> <p>Supported: TRUE   FALSE</p> <p>Endpoint(s): <a href="#">Get Sleep Logs</a>, <a href="#">Get Sleep Logs by Date Range</a></p>
level	string	<p>Returns the sleep stage for the timestamp provided.</p> <p>Supported: <a href="#">ASLEEP</a>   <a href="#">AWAKE</a>   <a href="#">RESTLESS</a>   <a href="#">WAKE</a>   <a href="#">LIGHT</a>   <a href="#">DEEP</a>   <a href="#">REM</a></p> <p>Endpoint(s): <a href="#">Get Sleep Logs</a>, <a href="#">Get Sleep Logs by Date Range</a>, <a href="#">Get Sleep Logs List</a></p>
minDuration	integer	<p>Time in minutes to achieve a sleep goal.</p> <p>Endpoint(s): <a href="#">Get Sleep Goal</a></p>
minutes	integer	<p>Total time in minutes spent in the associated level of sleep.</p> <p>Endpoint(s): <a href="#">Get Sleep Logs</a>, <a href="#">Get Sleep Logs by Date Range</a>, <a href="#">Get Sleep Logs List</a></p>
minutesAfterWakeup	integer	<p>The number of minutes the user was awake after being asleep</p>



		Endpoint(s): <a href="#">Get Sleep Logs</a> , <a href="#">Get Sleep Logs by Date Range</a> , <a href="#">Get Sleep Logs List</a>
minutesAsleep	integer	Number of minutes spent in deep, rem, light or asleep stages.  Endpoint(s): <a href="#">Get Sleep Logs</a> , <a href="#">Get Sleep Logs by Date Range</a> , <a href="#">Get Sleep Logs List</a>
minutesAwake	integer	Number of minutes spent in wake or awake stages.  Endpoint(s): <a href="#">Get Sleep Logs</a> , <a href="#">Get Sleep Logs by Date Range</a> , <a href="#">Get Sleep Logs List</a>
minutesToFallAsleep	integer	Number of minutes it took the user to fall asleep. This value will almost always be 0.  Endpoint(s): <a href="#">Get Sleep Logs</a> , <a href="#">Get Sleep Logs by Date Range</a> , <a href="#">Get Sleep Logs List</a>
seconds	integer	Number of seconds spent in the sleep stage.  Endpoint(s): <a href="#">Get Sleep Logs</a> , <a href="#">Get Sleep Logs by Date Range</a> , <a href="#">Get Sleep Logs List</a>
sleep : logType	string	Method of which the sleep was logged.  Supported: auto_detected   manual  Endpoint(s): <a href="#">Get Sleep Logs</a> , <a href="#">Get Sleep Logs by Date Range</a> , <a href="#">Get Sleep Logs List</a>
startTime	dateTime	Timestamp when the sleep log began; returned in <b>YYYY-MM-DDTHH:MM:SS:FFF</b> .  Endpoint(s): <a href="#">Get Sleep Logs</a> , <a href="#">Get Sleep Logs by Date Range</a> , <a href="#">Get Sleep Logs List</a>
thirtyDayAvgMinutes	integer	The minute average for the level of sleep in the last 30 days. This is only returned when type = "stages"  Endpoint(s): <a href="#">Get Sleep Logs</a> , <a href="#">Get Sleep Logs by Date Range</a> , <a href="#">Get Sleep Logs List</a>
timeInBed	integer	Number of minutes spent in bed.



		<p>Calculation: <math>\text{timeInBed} = \text{minutesAfterWakeup} + \text{minutesAsleep} + \text{minutesAwake} + \text{minutesToFallAsleep}</math></p> <p>Endpoint(s): <a href="#">Get Sleep Logs</a>, <a href="#">Get Sleep Logs by Date Range</a>, <a href="#">Get Sleep Logs List</a></p>
totalMinutesAsleep	integer	<p>Summation of the minutesAsleep values for all sleep logs recorded in the 24 hour period.</p> <p>Endpoint(s): <a href="#">Get Sleep Logs</a></p>
totalSleepRecords	integer	<p>Total number of sleep logs recorded in the 24 hour period</p> <p>Endpoint(s): <a href="#">Get Sleep Logs</a></p>
totalTimeInBed		<p>Summation of the timeInBed values for all sleep logs recorded in the 24 hour period.</p> <p>Endpoint(s): <a href="#">Get Sleep Logs</a></p>
type	string	<p>Type of sleep recorded. Supported: <a href="#">STAGES</a>   <a href="#">CLASSIC</a></p> <p>Classic sleep logs are generated when there is less than 3 hours of sleep stage data.</p> <p>Endpoint(s): <a href="#">Get Sleep Logs</a>, <a href="#">Get Sleep Logs by Date Range</a>, <a href="#">Get Sleep Logs List</a></p>
updatedOn	dateTime	<p>Timestamp of when the sleep goal was entered; returns in the format <b>YYYY-DD-MMTHH:MM:SS.FFF</b></p> <p>Endpoint(s): <a href="#">Get Sleep Goal</a></p>



## SpO2

The following information is returned from the SpO2 endpoints, logged by Fitbit devices worn by Fitbit users. [SpO2](#) data applies specifically to a user's blood oxygen level. The data returned include the minimum, maximum, and average percentage values of SpO2 in your bloodstream and timestamp for that measurement.

### GET SpO2

Element Name	Datatype	Description
dateTime	date	Date of log; in the format <b>YYYY-MM-DD</b> .  Endpoint(s): <a href="#">Get SpO2 Summary by Date</a> , <a href="#">Get SpO2 Summary by Interval</a>
value : avg	float	The mean of the 1 minute <a href="#">SpO2</a> levels calculated as a percentage value.  Endpoint(s): <a href="#">Get SpO2 Summary by Date</a> , <a href="#">Get SpO2 Summary by Interval</a>
value : min	float	The minimum daily <a href="#">SpO2</a> level calculated as a percentage value.  Endpoint(s): <a href="#">Get SpO2 Summary by Date</a> , <a href="#">Get SpO2 Summary by Interval</a>
value : max	float	The maximum daily <a href="#">SpO2</a> level calculated as a percentage value.  Endpoint(s): <a href="#">Get SpO2 Summary by Date</a> , <a href="#">Get SpO2 Summary by Interval</a>

### GET SpO2 Intraday Time Series

The data returned includes the percentage value of [SpO2](#) in your bloodstream and an accurate timestamp for that measurement.

Intraday data is only available by [request](#) and approved on a case by case basis.



Element Name	Datatype	Description
dateTime	date	Date of log; in the format <b>YYYY-MM-DD</b> .  Endpoint(s): <a href="#">Get SpO2 Intraday by Date</a> , <a href="#">Get SpO2 Intraday by Interval</a>
minutes : value	float	The percentage value of <a href="#">SpO2</a> calculated at a specific date and time in a single day.  Endpoint(s): <a href="#">Get SpO2 Intraday by Date</a> , <a href="#">Get SpO2 Intraday by Interval</a>
minutes : minute	dateTime	The date and time ( ) at which the <a href="#">SpO2</a> measurement was taken.  Endpoint(s): <a href="#">Get SpO2 Intraday by Date</a> , <a href="#">Get SpO2 Intraday by Interval</a>

## Temperature

The Temperature endpoints are used for querying either the core temperature data logged manually by the user, or the skin temperature recorded by the device while the user is asleep.

### GET Temperature

Element Name	Datatype	Description
tempCore : dateTime	dateTime	The log timestamp specified in the format <b>YYYY-MM-DDThh:mm:ss</b> .  Endpoint(s): <a href="#">Get Temperature (Core) Summary by Date</a> , <a href="#">Get Temperature (Core) Summary by Interval</a>
tempCore : value	integer	The temperature value is degrees Celsius or Fahrenheit depending on the country specified in the Accept-Language header.  Endpoint(s): <a href="#">Get Temperature (Core) Summary by Date</a> , <a href="#">Get Temperature (Core) Summary by Interval</a>



tempSkin : dateTime	dateTime	Date of the requested resource; in the format <b>YYYY-MM-DD</b> .  Endpoint(s): <a href="#">Get Temperature (Skin) Summary by Date</a> , <a href="#">Get Temperature (Skin) Summary by Interval</a>
tempSkin : value : nightlyRelative	integer	The user's average temperature during a period of sleep.  It is displayed to the user as a delta from their baseline temperature in degrees Celsius or Fahrenheit depending on the country specified in the Accept-Language header. See <a href="#">How can Fitbit help me track my temperature?</a> for more information.  Endpoint(s): <a href="#">Get Temperature (Skin) Summary by Date</a> , <a href="#">Get Temperature (Skin) Summary by Interval</a>
tempSkin : logType	string	The type of skin temperature log created. See <a href="#">Temperature Sensors</a> for more information.  Supported: dedicated_temp_sensor   other_sensors  Endpoint(s): <a href="#">Get Temperature (Skin) Summary by Date</a> , <a href="#">Get Temperature (Skin) Summary by Interval</a>



## Glossary

- **Activity Intensity Levels:** Stages of activity ranging from 0 - 6+ Metabolic Equivalents (MET). See: [https://en.wikipedia.org/wiki/Metabolic\\_equivalent\\_of\\_task](https://en.wikipedia.org/wiki/Metabolic_equivalent_of_task)
  - **Sedentary:** Activities with an MET value of less than 1.5
  - **Lightly Active:** Activities with an MET value of 1.5 - 3.0
  - **Moderately/Fairly Active:** Activities with an MET value of 3.0 - 6.0
  - **Very Active:** Activities with an MET value greater than 6.0
- **BMR:** Basal metabolic rate; or the rate at which you burn calories at rest to maintain vital body functions. See: [https://en.wikipedia.org/wiki/Basal\\_metabolic\\_rate](https://en.wikipedia.org/wiki/Basal_metabolic_rate)
- **BPM:** Beats per minute
- **Classic:** Short for Classic Sleep. Levels data returned with 60-second granularity. 'Sleep Pattern' levels include asleep, restless, and awake.
- **Classification of ECG**
  - **Atrial Fibrillation (AFib):** Your heart rhythm shows signs of AFib, an irregular heart rhythm. AFib can have serious health effects. You should contact your doctor.
  - **Inconclusive:** If your heart rate is over 120 bpm or under 50 bpm, the Fitbit ECG app can't assess your heart rhythm. There are many possible reasons for getting an inconclusive result, but common causes are moving too much during the assessment, not resting your hands on a table, or other arrhythmia.
  - **Inconclusive: High Heart Rate:** If your heart rate is over 120 bpm, the Fitbit ECG app can't assess your heart rhythm. Heart rate can be high for many reasons, such as: recent exercise, stress, nervousness, alcohol, dehydration, infection, AFib, or other arrhythmia.
  - **Inconclusive: Low Heart Rate:** If your heart rate is under 50 bpm, the Fitbit ECG app can't assess your heart rhythm. Heart rate can be low for many reasons, such as: taking certain medications such as beta-blockers or calcium channel blockers, having excellent aerobic fitness, or other arrhythmia.
  - **Normal Sinus Rhythm (NSR):** Your heart rhythm appears normal. It doesn't show signs of AFib, an irregular heart rhythm.
- Food plan intensity levels are defined below.
  - **Easier:** Lose 0.5 lbs per week with a rate of -250 calories per day
  - **Medium:** Lose 1 lb per week with a rate of -500 calories per day
  - **Kinda Hard:** Lose 1.5 lbs per week with a rate of -750 calories per day
- **Harder:** Lose 2 lbs a week with a rate of -1,000 calories per day **Heart rate zones** - include Fat Burn, Cardio, Peak and Out of Range
  - **Out of Range:** Below 50% of the user's maximum heart rate
  - **Fat Burn:** Between 50% - 69% of the user's maximum heart rate
  - **Cardio:** Between 70% - 84% of the user's maximum heart rate



- **Peak:** Greater than 80% of the user's maximum heart rate
- **Intraday:** More granular level of Activity and Heart Rate time series data returned within a 24 hour period
- **Lap:** A repetitive distance traveled within an exercise. This is defined by the user.
- **Maximum Heart Rate:** 220 minus age in years
- **Oxygen Saturation (SpO2):** A measure of the amount of oxygen-carrying hemoglobin in the blood relative to the amount of hemoglobin not carrying oxygen.
- **Root Mean Square of Successive Differences (RMSSD):** reflects the beat-to-beat variance in HR.
- **Stages:** Short for Sleep Stages. Levels data is returned with 30-second granularity. 'Sleep Stages' levels include **deep**, **light**, **rem** (rapid eye movement), and **wake**.