Google Fitbit

Fitbit Web API Data Dictionary

Version 9; last updated Aug 19, 2024

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 sedentary level. This does not include calories burned from Basal metabolic rate (BMR).

		Endpoint(s): Get Daily Activity Summary
activeDuration	integer	The amount of time in each activityLevel. Time in milliseconds. Endpoint(s): Get Activity Logs List
activeMinutes	integer	User defined goal for daily active minutes. Endpoint(s): Get Daily Activity Summary
activeZoneMinutes	Integer	Daily or weekly active zone minutes goal. Endpoint(s): Get Activity Goals
activityId	integer	The recorded exercise's identifier number. For example, the activityId for “Run” is 90009. For a complete list of exercise identifiers, see Browse Activity Type endpoint. Endpoint(s): Get Daily Activity Summary
activityLevel	integer	Returns minutes spent in each activity zone: sedentary, lightly active, moderately active, and very active. Endpoint(s): Get Activity Logs List
activityName	string	Name of the recorded exercise (i.e. Walk, Run, Elliptical, Hike, and custom exercises) Endpoint(s): Get Activity Logs List
altitudemeters	integer	The altitude meters recorded at the specified time. Endpoint(s): Get Activity TCX
averageHeartRate	integer	Average heart rate during the exercise from start to finish. Endpoint(s): Get Activity Logs List
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 recorded above sedentary level, excluding caloriesBMR.

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

		Endpoint(s): Get Activity TCX
distanceUnit	string	Distance units defined by the Accept-Language header. Endpoint(s): Get Activity Logs List
distances	list	Total distance accumulated for the day for the following elements: • total • tracker • loggedActivities • veryActive • moderatelyActive • lightlyActive • sedentaryActive Endpoint(s): Get Daily Activity Summary
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): Get Daily Activity Summary, Get Frequent Activities, Get Recent Activity Types
elevation	integer	The elevation traveled for the day displayed in the units defined by the Accept-Language header. Endpoint(s): Get Daily Activity Summary
elevationGain	integer	Elevation gained during the exercise. Endpoint(s): Get Activity Logs List
fairlyActiveMinutes	integer	Total minutes the user was fairly/moderately active. Endpoint(s): Get Daily Activity Summary
floors	integer	The equivalent floors climbed associated with the activity, goal, summary, and lifetime totals. Displayed in the units defined by the Accept-Language header. Endpoint(s): Get Daily Activity Summary, Get Activity Goals, Get Lifetime Stats

heartRateLink	string	Link to fetch the intraday heart rate data for the activity recorded, using the date and time the activity took place. Endpoint(s): Get Activity Logs List
heartRateZones	list	Returns the minutes spent in, and min/max values, of each heart rate zone for: • Out of range • Fat Burn • Cardio • Peak Endpoint(s): Get Activity Logs List
intensity	integer	Integer representing the intensity level of the activity. • 0 = Sedentary • 1 = Lightly active • 2 = Moderately/Fairly active • 3 = Very active Endpoint(s): Get Activity TCX
lap (starttime)	time	Timestamp representing each exercise lap start time. Endpoint(s): Get Activity TCX
latitudedegrees	integer	The GPS latitude of the recorded exercise at the specified time Endpoint(s): Get Activity TCX
longtitudedegrees	integer	The GPS longitude of the recorded exercise at the specified time. Endpoint(s): Get Activity TCX
lightlyActiveMinutes	integer	Total minutes the user was lightly active. Endpoint(s): Get Daily Activity Summary
logType	string	Method of which the activity was logged. Supported: manual | mobile_run | tracker | auto_detected | fitstar | or the <name> of the 3rd party application

		Endpoint(s): Get Activity Logs List
manualValuesSpecified		Manually logged counts for the following elements: calories, distance, steps. Endpoint(s): Get Activity Logs List
marginalCalories	integer	Estimated marginal calories burned. Endpoint(s): Get Daily Activity Summary
mets	integer	The metabolic equivalent (METs) of the activity performed. Endpoint(s): Get Activity Type, Get Favorite Activities
minutes	integer	Total number of minutes the user spent during the specified activity level for that day. Endpoint(s): Get Activity Logs List
name	string	Name of the activity level. Supported: sedentary | lightly | fairly | very Endpoint(s): Get Activity Logs List
name	string	Name of the recorded exercise. Endpoint(s): Get Daily Activity Summary
originalDuration	integer	The initial length in time (milliseconds) that the exercise was recorded. This value will contain pauses during the exercise. Endpoint(s): Get Activity Logs List
originalStartTime	integer	The initial start datetime that the exercise was recorded. Endpoint(s): Get Activity Logs List
pace	integer	Calculated average pace during the exercise. Endpoint(s): Get Activity Logs List
sedentaryMinutes	integer	Total minutes the user was sedentary.

		Endpoint(s): Get Daily Activity Summary
speed	integer	Average speed during the exercise. Endpoint(s): Get Activity Logs List
startDate	date	The start date of the recorded exercise. Endpoint(s): Get Daily Activity Summary
startTime	date/dateTime	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): Get Daily Activity Summary, Get Activity Logs List
steps	integer	Step counts associated with the activity, goal, summary, and lifetime totals. Endpoint(s): Get Daily Activity Summary, Get Activity Logs List, Get Activity Goals, Get Lifetime Stats
time	time	The time that metrics were recorded during the exercise. Endpoint(s): Get Activity TCX
totaltimeseconds	integer	Length of the exercise lap in seconds Endpoint(s): Get Activity TCX
value	integer	The heart rate recorded at the specified timestamp. Endpoint(s): Get Activity TCX
veryActiveMinutes	integer	Total minutes the user was very active. Endpoint(s): Get Daily Activity Summary

GET Activity Time Series

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

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	Resource can be one of the following values:

		Calories | Steps | Distance | Floors | Elevation | activeZoneMinutes The selected resource will be returned with the summary value for the given date and time range. Endpoint(s): Get Activity Intraday Timeseries
dateTime	date	Date of the requested resource; in the format YYYY-MM-DD. Endpoint(s): Get Activity Intraday Timeseries
level	integer	Numerical value representing the user's activity-level at the moment when the resource was recorded. 0 = sedentary 1 = lightly active 2 = fairly/moderately active 3 = very active Returned only when resource = calories Endpoint(s): Get Activity Intraday Timeseries
mets	integer	METs value at the moment when the resource was recorded. Returned only when resource = calories Endpoint(s): Get Activity Intraday Timeseries
time	time	The time the resource was recorded; in the format HH:MM Endpoint(s): Get Activity Intraday Timeseries
value	integer	The specified resource's value at the time it is recorded Endpoint(s): Get Activity Intraday Timeseries

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 YYYY-MM-DD or today. Endpoint(s): Get AZM Daily Summary by Date, Get AZM Daily Summary by Interval
activeZoneMinutes	integer	Total count of active zone minutes Endpoint(s): Get AZM Daily Summary by Date, Get AZM Daily Summary by Interval
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): Get AZM Daily Summary by Date, Get AZM Daily Summary by Interval
cardioActiveZoneMinutes	integer	The number of active zone minutes in the cardio heart rate zone. 1 cardio minute = 2 cardio active zone minutes. Endpoint(s): Get AZM Daily Summary by Date, Get AZM Daily Summary by Interval
peakActiveZoneMinutes	integer	The number of active zone minutes in the peak heart rate zone. 1 peak minute = 2 peak active zone minutes. Endpoint(s): Get AZM Daily Summary by Date, Get AZM Daily Summary by Interval

GET AZM Intraday Time Series

Element Name	Datatype	Description
dateTime	date	Date of log; in the format YYYY-MM-DD or today. Endpoint(s): Get AZM Intraday by Date, Get AZM Intraday by Interval
minute	dateTime	The dateTime when the AZM value was recorded. Endpoint(s): Get AZM Intraday by Date, Get AZM Intraday by Interval
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): Get AZM Intraday by Date, Get AZM Intraday by Interval
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): Get AZM Intraday by Date, Get AZM Intraday by Interval
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): Get AZM Intraday by Date, Get AZM Intraday by Interval
activeZoneMinutes	integer	Total count of active zone minutes earned during the previous iteration. Endpoint(s): Get AZM Intraday by Date, Get AZM Intraday by Interval

Body & Weight

GET Body & Weight

Element Name	Datatype	Description

bmi	float	Calculated BMI in the format X.XX. Body mass index is a value derived from the mass and height of a person. Endpoint(s): Get Weight Logs
date	date	The date in the format yyyy-MM-dd. Endpoint(s): Get Body Fat Logs, Get Weight Logs
fat	float	Body fat percentage; in the format X.XX. • Returned only when goal type is set to fat.* Endpoint(s): Get Body Fat Logs, Get Body Goals*, Get Weight Logs
source	string	The source of the weight log. Supported: API | Aria | AriaAir | Withings Endpoint(s): Get Body Fat Logs, Get Weight Logs
startDate	date	The start date of the body goal; in the format YYYY-MM-DD. • Returned only when goal type is set to weight. Endpoint(s): Get Body Goals
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. • Returned only when goal type is set to weight. Endpoint(s): Get Body Goals
time	time	Time of the measurement; hours and minutes in the format HH:mm:ss, set to the last second of the day if not provided. Endpoint(s): Get Body Fat Logs, Get Weight Logs
weight	float	Weight in the format X.XX, in the unit system that corresponds to the Accept-Language header provided or if not provided in metric.

		• Returned only when goal type is set to weight.* Endpoint(s): Get Body Goals*, Get Weight Logs

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): Get Body Time Series
dateTime	date	Date of the requested resource; in the format YYYY-MM-DD. Endpoint(s): Get Body Time Series
value	float	The value recorded at the specific timestamp. Endpoint(s): Get Body Time Series

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 YYYY-MM-DD.

		Endpoint(s): Get Breathing Rate Summary by Date, Get Breathing Rate Summary by Interval
br : value : breathingRate	float	Average number of breaths taken per minute. Endpoint(s): Get Breathing Rate Summary by Date, Get Breathing Rate Summary by Interval

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 YYYY-MM-DD. Endpoint(s): Get Breathing Rate Intraday by Date, Get Breathing Rate Intraday by Interval
br : value : lightSleepSummary : breathingRate	float	Average number of breaths taken per minute when the user was in light sleep. Endpoint(s): Get Breathing Rate Intraday by Date, Get Breathing Rate Intraday by Interval
br : value : deepSleepSummary : breathingRate	float	Average number of breaths taken per minute when the user was in deep sleep. Endpoint(s): Get Breathing Rate Intraday by Date, Get Breathing Rate Intraday by Interval
br : value : remSleepSummary : breathingRate	integer	Average number of breaths taken per minute when the user was in REM sleep. Endpoint(s): Get Breathing Rate Intraday by Date, Get Breathing Rate Intraday by Interval
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): Get Breathing Rate Intraday by Date, Get Breathing Rate Intraday by Interval

Cardio Fitness Score (VO2 Max)

GET VO2 Max Summary

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

Devices

GET Devices

battery	integer	Numerical value representing the percentage of the device's battery life. Supported: 0-100 Endpoint(s): Get Devices
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): Get Devices
enabled	boolean	true or false; if false, alarm does not vibrate until enabled is set to true Supported: True | False

		Endpoint(s): Get Alarms
features	string	Placeholder for displaying the device features. At this time, an empty string is returned. Endpoint(s): Get Devices
lastSyncTime	date	The time the device last synced with the Fitbit mobile app; returned in the format YYYY-DD-MMTHH:MM:SS:FFF Endpoint(s): Get Devices
recurring	boolean	true or false; if false, the alarm is a single event. Supported: True | False Endpoint(s): Get Alarms
snoozeCount	integer	Maximum number of times a user can snooze the alarm. Endpoint(s): Get Alarms
snoozeLength	integer	Number of minutes until the next snooze alarm occurs. Endpoint(s): Get Alarms
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): Get Alarms
time	time	Time of day that the alarm vibrates with a UTC timezone offset, e.g. 07:15-08:00 Endpoint(s): Get Alarms
type	string	Type of device Supported: Tracker | Scale Endpoint(s): Get Devices

vibe	string	Vibration pattern Supported: DEFAULT Endpoint(s): Get Alarms
weekDays	list	List of days of the week on which the alarm vibrates Supported: Sunday | Monday | Tuesday | Wednesday | Thursday | Friday | Saturday Endpoint(s): Get Alarms

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): Get ECG Log List
averageHeartRate	integer	The average heart rate of the user during the recording. Endpoint(s): Get ECG Log List
beforeDate	dateTime	The beforeDate parameter of the request. Endpoint(s): Get ECG Log List
deviceName	string	Hardware name of the device used to take the measurement. Endpoint(s): Get ECG Log List
leadNumber	integer	The ECG lead being used to take the reading. For the

		Fitbit devices, the leadNumber will always be 1. Supported: 1 Endpoint(s): Get ECG Log List
numberOfWaveformSamples	integer	The total number of samples in the recording. Endpoint(s): Get ECG Log List
previous	string	The URL of the request that will fetch the previous page of results. Endpoint(s): Get ECG Log List
resultClassification	string	See Classification of ECG results for more information. Supported: Atrial Fibrillation | Normal Sinus Rhythm | Inconclusive | Inconclusive: High heart rate | Inconclusive: Low heart rate Endpoint(s): Get ECG Log List
startTime	dateTime	The date and time when the reading was started on the device. Endpoint(s): Get ECG Log List
samplingFrequencyHz	integer	The frequency in hertz at which Fitbit sampled the voltage. Supported: 250 Endpoint(s): Get ECG Log List
scalingFactor	integer	The scaling factor used to convert waveform samples to ECG voltages in mV (mV = sample / scalingFactor). Supported: 10922 Endpoint(s): Get ECG Log List
waveFormSamples	array	An array of integers representing the ECG waveform visible to the user on their PDF report. Endpoint(s): Get ECG Log List

Food Logging

GET Food & Water

amount	integer	Quantity of the food or water entry displayed in the units defined by the Accept-Language header. Endpoint(s): Get Food Logs, Get Water Logs, Get Frequent Foods, Get Recent Foods
barcode	boolean	Value determining if barcode was provided. Supported: TRUE | FALSE Endpoint(s): Get Food Locales
brand	string	Brand of the food logged Endpoint(s): Get Food Logs, Get Favorite Foods, Get Frequent Foods, Get Recent Foods, Get Food
carbs	float	Amount of carbohydrates in the recorded food entry; returned in grams. For a complete list... Endpoint(s): Get Food Logs, Get Favorite Foods, Get Food
calories	integer	Value for daily calorie consumption. Endpoint(s): Get Food Goals, Get Food Logs, Get Favorite Foods, Get Frequent Foods, Get Recent Foods, Get Food, Get Search Foods
dateLastEaten	date	Date of when the food was last consumed. Endpoint(s): Get Frequent Foods
defaultServingSize	integer	The default serving size associated with the food log. Endpoint(s): Get Favorite Foods, Get Food, Get Search Foods
defaultUnit	list	Returns information associated to the unit specified:

		• id: numerical identifier for the unit • name: name of the unit • plural: abbreviation of the unit Endpoint(s): Get Favorite Foods, Get Food, Get Search Foods
estimatedCaloriesOut	integer	Number of calories expected to burn from BMR plus the calories needed to burn through activity based on the food plan intensity goal. Example: If BMR calories = 800 and Food Plan intensity is "Harder" (1000 calories), estimated calories out is 800 + 1000 = 1800 Endpoint(s): Get Food Logs
estimatedDate	date	Estimated date of goal completion; in the format yyyy-MM-dd. • Returned only when intensity type is set to one of the following: EASIER | MEDIUM | KINDA HARD | HARDER Endpoint(s): Get Food Goals
fat	float	Amount of fat in the recorded food entry; returned in grams. Endpoint(s): Get Food Logs, Get Favorite Foods, Get Food
fiber	float	Amount of fiber in the recorded food entry; returned in grams. Endpoint(s): Get Food Logs, Get Favorite Foods, Get Food
goal	integer	Amount of water to be consumed set by the user's goal; returned in milliliters. Endpoint(s): Get Water Goal
intensity	string	Level of difficulty for the selected food plan. Supported: MAINTENANCE | EASIER | MEDIUM | KINDA HARD | HARDER

		Endpoint(s): Get Food Goals
isFavorite	boolean	Value determining if the selected food is categorized as one of the user's favorite foods. Supported: TRUE | FALSE Endpoint(s): Get Food Logs
mealTypeId	integer	Numerical value associated with when the food was consumed. • 1=Breakfast • 2=Morning Snack • 3=Lunch • 4=Afternoon Snack • 5=Dinner • 7=Anytime. Endpoint(s): Get Frequent Foods
name	string	Name of the food. Endpoint(s): Get Food Logs, Get Frequent Foods, Get Recent Foods, Get Food, Get Search Foods, Get Favorite Foods
personalized	boolean	Specifies if the user personalized the selected food plan. Supported: TRUE | FALSE Endpoint(s): Get Food Goals
protein	float	Amount of protein in the recorded food entry; returned in grams. Endpoint(s): Get Food Logs, Get Favorite Foods, Get Food
servings	list	Metadata related to the food's servings: • Multiplier: Number of times the serving size was consumed. • servingSize: The number of servings within the food product.

		Endpoint(s): Get Favorite Foods, Get Food
sodium	float	Amount of Sodium in the recorded food entry; returned in grams. Endpoint(s): Get Food Logs, Get Favorite Foods, Get Food
startDate	date	Start date of the water goal entry. Endpoint(s): Get Water Goal
water	float	Daily water consumption displayed in the units defined by the Accept-Language header. Endpoint(s): Get Water Logs, Get Food Logs

GET Food & Water Time Series

dateTime	date	Date of the requested resource; in the format YYYY-MM-DD. Endpoint(s): Get Food or Water Timeseries
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): Get Food or Water Timeseries
value	integer	Total count of the requested resource. Endpoint(s): Get Food or Water Timeseries

Nutritional Values

biotin	integer	Amount of biotin in the recorded food entry; returned

		in grams. Endpoint(s): Get Food Logs, Get Favorite Foods, Get Food
calcium	integer	Amount of calcium in the recorded food entry; returned in grams. Endpoint(s): Get Food Logs, Get Favorite Foods, Get Food
calories	integer	Value for daily calorie consumption. Endpoint(s): Get Food Goals, Get Food Logs, Get Favorite Foods, Get Frequent Foods, Get Recent Foods, Get Food, Get Search Foods
caloriesFromFat	integer	Amount of calories from fat in the recorded food entry; returned in grams. Endpoint(s): Get Food Logs, Get Favorite Foods, Get Food
cholesterol	integer	Amount of cholesterol in the recorded food entry; returned in grams. Endpoint(s): Get Food Logs, Get Favorite Foods, Get Food
copper	integer	Amount of copper in the recorded food entry; returned in grams. Endpoint(s): Get Food Logs, Get Favorite Foods, Get Food
dietaryFiber	integer	Amount of dietaryFiber in the recorded food entry; returned in grams. Endpoint(s): Get Food Logs, Get Favorite Foods, Get Food
fat	float	Amount of fat in the recorded food entry; returned in grams. Endpoint(s): Get Food Logs, Get Favorite Foods, Get Food
fiber	float	Amount of fiber in the recorded food entry; returned

		in grams. Endpoint(s): Get Food Logs, Get Favorite Foods, Get Food
folicAcid	integer	Amount of folicAcid in the recorded food entry; returned in grams. Endpoint(s): Get Food Logs, Get Favorite Foods, Get Food
iodine	integer	Amount of iodine in the recorded food entry; returned in grams. Endpoint(s): Get Food Logs, Get Favorite Foods, Get Food
iron	integer	Amount of iron in the recorded food entry; returned in grams. Endpoint(s): Get Food Logs, Get Favorite Foods, Get Food
magnesium	integer	Amount of magnesium in the recorded food entry; returned in grams. Endpoint(s): Get Food Logs, Get Favorite Foods, Get Food
niacin	integer	Amount of niacin in the recorded food entry; returned in grams. Endpoint(s): Get Food Logs, Get Favorite Foods, Get Food
pantothenicAcid	integer	Amount of pantothenicAcid in the recorded food entry; returned in grams. Endpoint(s): Get Food Logs, Get Favorite Foods, Get Food
phosphorus	integer	Amount of phosphorus in the recorded food entry; returned in grams. Endpoint(s): Get Food Logs, Get Favorite Foods, Get Food
potassium	integer	Amount of potassium in the recorded food entry;

		returned in grams. Endpoint(s): Get Food Logs, Get Favorite Foods, Get Food
protein	float	Amount of protein in the recorded food entry; returned in grams. Endpoint(s): Get Food Logs, Get Favorite Foods, Get Food
riboflavin	integer	Amount of riboflavin in the recorded food entry; returned in grams. Endpoint(s): Get Food Logs, Get Favorite Foods, Get Food
saturatedFat	integer	Amount of saturated fat in the recorded food entry; returned in grams. Endpoint(s): Get Food Logs, Get Favorite Foods, Get Food
sodium	float	Amount of Sodium in the recorded food entry; returned in grams. Endpoint(s): Get Food Logs, Get Favorite Foods, Get Food
sugars	integer	Amount of sugars in the recorded food entry; returned in grams. Endpoint(s): Get Food Logs, Get Favorite Foods, Get Food
thiamin	integer	Amount of thiamin in the recorded food entry; returned in grams. Endpoint(s): Get Food Logs, Get Favorite Foods, Get Food
totalCarbohydrate	integer	Amount of total carbohydrates in the recorded food entry; returned in grams. Endpoint(s): Get Food Logs, Get Favorite Foods, Get Food
totalFat	integer	Amount of total fat in the recorded food entry;

		returned in grams. Endpoint(s): Get Food Logs, Get Favorite Foods, Get Food
transFat	integer	Amount of trans fat in the recorded food entry; returned in grams. Endpoint(s): Get Food Logs, Get Favorite Foods, Get Food
vitaminA	integer	Amount of vitamin A in the recorded food entry; returned in grams. Endpoint(s): Get Food Logs, Get Favorite Foods, Get Food
vitaminB12	integer	Amount of vitamin B12 in the recorded food entry; returned in grams. Endpoint(s): Get Food Logs, Get Favorite Foods, Get Food
vitaminB6	integer	Amount of vitamin B6 in the recorded food entry; returned in grams. Endpoint(s): Get Food Logs, Get Favorite Foods, Get Food
vitaminC	integer	Amount of vitamin C in the recorded food entry; returned in grams. Endpoint(s): Get Food Logs, Get Favorite Foods, Get Food
vitaminD	integer	Amount of vitamin D in the recorded food entry; returned in grams. Endpoint(s): Get Food Logs, Get Favorite Foods, Get Food
vitaminE	integer	Amount of vitamin E in the recorded food entry; returned in grams. Endpoint(s): Get Food Logs, Get Favorite Foods, Get Food
zinc	integer	Amount of zinc in the recorded food entry; returned

		in grams. Endpoint(s): Get Food Logs, Get Favorite Foods, Get Food

Heart

GET Heart Rate Time Series

caloriesOut	integer	Number of calories burned with custom and non-custom heart rate zones. Endpoint(s): Get Heart Rate Time Series
dateTime	date	Date of the heart rate log; returned in the format YYYY-DD-MM. Endpoint(s): Get Heart Rate Time Series
max	integer	Maximum heart rate range value for custom and non-custom heart rate zones. Endpoint(s): Get Heart Rate Time Series
min	integer	Minimum heart rate range value for custom and non-custom heart rate zones. Endpoint(s): Get Heart Rate Time Series
minutes	integer	Number of minutes within custom and non-custom heart rate zones. Endpoint(s): Get Heart Rate Time Series
name	string	Name of the custom and non-custom heart rate zones. Endpoint(s): Get Heart Rate Time Series
restingHeartRate	integer	BPM value while at complete rest. RHR uses heart-rate data from both awake and asleep states to estimate RHR.

		Endpoint(s): Get Heart Rate Time Series

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 custom and non-custom heart rate zones. Endpoint(s): Get Heart Rate Intraday Time Series
dateTime	date	Date of the heart rate log; returned in the format YYYY-MM-DD. Endpoint(s): Get Heart Rate Intraday Time Series
max	integer	Maximum heart rate range value for custom and non-custom heart rate zones. Endpoint(s): Get Heart Rate Time Series, Get Heart Rate Intraday Time Series
min	integer	Minimum heart rate range value for custom and non-custom heart rate zones. Endpoint(s): Get Heart Rate Intraday Time Series
minutes	integer	Number of minutes within the custom and non-custom heart rate zones. Endpoint(s): Get Heart Rate Intraday Time Series
name	string	Name of the custom and non-custom heart rate zones. Endpoint(s): Get Heart Rate Intraday Time Series
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): Get Heart Rate Intraday Time Series

time	time	The time the intraday heart rate value was recorded; returned in the format HH:MM:SS. Endpoint(s): Get Heart Rate Intraday Time Series
value	integer	This element will represent one of the following: 1. The heart rate value at the time the reading was recorded. 2. The average sum of the heart rate values in the activities-heart-intraday dataset. It will return only when start-time and end-time are specified in the endpoint. Endpoint(s): Get Heart Rate Intraday Time Series

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 YYYY-MM-DD. Endpoint(s): Get HRV Summary by Date, Get HRV Summary by Interval
value : dailyRmssd	float	The Root Mean Square of Successive Differences (RMSSD) between heart beats. It measures short-term variability in the user's daily heart rate in milliseconds (ms). Endpoint(s): Get HRV Summary by Date, Get HRV Summary by Interval
value : deepRmssd	float	The Root Mean Square of Successive Differences (RMSSD) between heart beats. It measures short-term variability in the user's heart rate while in deep sleep, in milliseconds (ms). Endpoint(s): Get HRV Summary by Date, Get HRV Summary by Interval

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

Irregular Rhythm Notifications (IRN)

The Irregular Rhythm Notifications (also known as IRN) endpoints are used for querying the user's engagement with the IRN feature and the alerts received.

alertTime	dateTime	The start time for the irregular rhythm detection, returned in the format **yyyy-MM-ddTHH:mm:ss**. Endpoint(s): Get IRN Alerts List
detectedTime	dateTime	The end time for the irregular rhythm detection, returned in the format **yyyy-MM-ddTHH:mm:ss**. Endpoint(s): Get IRN Alerts List
deviceType	string	The name of the device who generated the alert. Endpoint(s): Get IRN Alerts List
startTime	dateTime	The start time for the analyzable window (representing 5 consecutive minutes of data following the start time), returned in the format **yyyy-MM-ddTHH:mm:ss**. Endpoint(s): Get IRN Alerts List
dataTime	dateTime	The timestamp of the individual heart beat, returned in the format **yyyy-MM-ddTHH:mm:ss**. Endpoint(s): Get IRN Alerts List
bpmData : value	integer	The extrapolated bpm value from the individual heart beat. Endpoint(s): Get IRN Alerts List
onboarded	boolean	Whether or not the user has on-boarded onto the IRN feature. Endpoint(s): Get IRN Profile
enrolled	boolean	Whether or not the user is currently enrolled in having their data processed for IRN alerts. Endpoint(s): Get IRN Profile

lastUpdated	dateTime	The timestamp of the last piece of analyzable data synced by the user (displayed as local time), returned in the format **yyyy-MM-ddTHH:mm:ss**. Endpoint(s): Get IRN Profile

Sleep

GET Sleep Logs

count	integer	Total count of how many times a user was in the associated sleep stage. Endpoint(s): Get Sleep Logs, Get Sleep Logs by Date Range, Get Sleep Logs List
dateOfSleep	date	Date of recorded sleep; returned in YYYY-DD-MM. Endpoint(s): Get Sleep Logs, Get Sleep Logs by Date Range, Get Sleep Logs List
dateTime	dateTime	Timestamp when the user enters the sleep stage Endpoint(s): Get Sleep Logs, Get Sleep Logs by Date Range, Get Sleep Logs List
duration	integer	Duration of sleep log in milliseconds. Endpoint(s): Get Sleep Logs, Get Sleep Logs by Date Range, Get Sleep Logs List
efficiency	integer	An algorithm based on collected sleep metrics to determine how efficient the user slept. Endpoint(s): Get Sleep Logs, Get Sleep Logs by Date Range, Get Sleep Logs List
endTime	dateTime	Timestamp when the sleep log ended; returned in YYYY-DD-MMTHH:MM:SS:FFF.

		Endpoint(s): Get Sleep Logs
flowId	integer	An integer value representing the sleep goal consistency flow. • 0 = A sleep goal is set, but there are not enough sleep logs recorded. • 1 = The user either missed their sleep goal or no goal is set, but there are enough sleep logs recorded. • 2 = A sleep goal is not set, and there are not enough sleep logs recorded. • 3 = The user achieved their sleep goal. Endpoint(s): Get Sleep Goals
infoCode	integer	An integer value representing the quality of data collected within the sleep log. • 0 = Sufficient data to generate a sleep log. • 1 = Insufficient heart rate data. • 2 = Sleep period was too short (less than 3 hours). • 3 = Server-side issue Endpoint(s): Get Sleep Logs List
isMainSleep	boolean	States if the sleep record was the longest sleep of the day. Supported: TRUE | FALSE Endpoint(s): Get Sleep Logs, Get Sleep Logs by Date Range
level	string	Returns the sleep stage for the timestamp provided. Supported: ASLEEP | AWAKE | RESTLESS | WAKE | LIGHT | DEEP | REM

		Endpoint(s): Get Sleep Logs, Get Sleep Logs by Date Range, Get Sleep Logs List
minDuration	integer	Time in minutes to achieve a sleep goal. Endpoint(s): Get Sleep Goal
minutes	integer	Total time in minutes spent in the associated level of sleep. Endpoint(s): Get Sleep Logs, Get Sleep Logs by Date Range, Get Sleep Logs List
minutesAfterWakeup	integer	The number of minutes the user was awake after being asleep Endpoint(s): Get Sleep Logs, Get Sleep Logs by Date Range, Get Sleep Logs List
minutesAsleep	integer	Number of minutes spent in deep, rem, light or asleep stages. Endpoint(s): Get Sleep Logs, Get Sleep Logs by Date Range, Get Sleep Logs List
minutesAwake	integer	Number of minutes spent in wake or awake stages. Endpoint(s): Get Sleep Logs, Get Sleep Logs by Date Range, Get Sleep Logs List
minutesToFallAsleep	integer	Number of minutes it took the user to fall asleep. This value will almost always be 0. Endpoint(s): Get Sleep Logs, Get Sleep Logs by Date Range, Get Sleep Logs List
seconds	integer	Number of seconds spent in the sleep stage. Endpoint(s): Get Sleep Logs, Get Sleep Logs by Date Range, Get Sleep Logs List
sleep : logType	string	Method of which the sleep was logged. Supported: auto_detected | manual Endpoint(s): Get Sleep Logs, Get Sleep Logs by Date Range, Get Sleep Logs List

startTime	dateTime	Timestamp when the sleep log began; returned in YYYY-MM-DDTHH:MM:SS:FFF. Endpoint(s): Get Sleep Logs, Get Sleep Logs by Date Range, Get Sleep Logs List
thirtyDayAvgMinutes	integer	The minute average for the level of sleep in the last 30 days. This is only returned when type = "stages" Endpoint(s): Get Sleep Logs, Get Sleep Logs by Date Range, Get Sleep Logs List
timeInBed	integer	Number of minutes spent in bed. Calculation: timeInBed = minutesAfterWakeup + minutesAsleep + minutesAwake + minutesToFallAsleep Endpoint(s): Get Sleep Logs, Get Sleep Logs by Date Range, Get Sleep Logs List
totalMinutesAsleep	integer	Summation of the minutesAsleep values for all sleep logs recorded in the 24 hour period. Endpoint(s): Get Sleep Logs
totalSleepRecords	integer	Total number of sleep logs recorded in the 24 hour period Endpoint(s): Get Sleep Logs
totalTimeInBed		Summation of the timeInBed values for all sleep logs recorded in the 24 hour period. Endpoint(s): Get Sleep Logs
type	string	Type of sleep recorded. Supported: STAGES | CLASSIC Classic sleep logs are generated when there is less than 3 hours of sleep stage data. Endpoint(s): Get Sleep Logs, Get Sleep Logs by Date Range, Get Sleep Logs List

updatedOn	dateTime	Timestamp of when the sleep goal was entered; returns in the format YYYY-DD-MMTHH:MM:SS.FFF Endpoint(s): Get Sleep Goal

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 YYYY-MM-DD. Endpoint(s): Get SpO2 Summary by Date, Get SpO2 Summary by Interval
value : avg	float	The mean of the 1 minute SpO2 levels calculated as a percentage value. Endpoint(s): Get SpO2 Summary by Date, Get SpO2 Summary by Interval
value : min	float	The minimum daily SpO2 level calculated as a percentage value. Endpoint(s): Get SpO2 Summary by Date, Get SpO2 Summary by Interval
value : max	float	The maximum daily SpO2 level calculated as a percentage value. Endpoint(s): Get SpO2 Summary by Date, Get SpO2 Summary by Interval

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 YYYY-MM-DD. Endpoint(s): Get SpO2 Intraday by Date, Get SpO2 Intraday by Interval
minutes : value	float	The percentage value of SpO2 calculated at a specific date and time in a single day. Endpoint(s): Get SpO2 Intraday by Date, Get SpO2 Intraday by Interval
minutes : minute	dateTime	The date and time () at which the SpO2 measurement was taken. Endpoint(s): Get SpO2 Intraday by Date, Get SpO2 Intraday by Interval

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 YYYY-MM-DDThh:mm:ss. Endpoint(s): Get Temperature (Core) Summary by Date, Get Temperature (Core) Summary by Interval

tempCore : value	integer	The temperature value is degrees Celsius or Fahrenheit depending on the country specified in the Accept-Language header. Endpoint(s): Get Temperature (Core) Summary by Date, Get Temperature (Core) Summary by Interval
tempSkin : dateTime	dateTime	Date of the requested resource; in the format YYYY-MM-DD. Endpoint(s): Get Temperature (Skin) Summary by Date, Get Temperature (Skin) Summary by Interval
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 How can Fitbit help me track my temperature? for more information. Endpoint(s): Get Temperature (Skin) Summary by Date, Get Temperature (Skin) Summary by Interval
tempSkin : logType	string	The type of skin temperature log created. See Temperature Sensors for more information. Supported: dedicated_temp_sensor | other_sensors Endpoint(s): Get Temperature (Skin) Summary by Date, Get Temperature (Skin) Summary by Interval

Glossary

- Activity Intensity Levels: Stages of activity ranging from 0 - 6+ Metabolic Equivalents (MET). See: 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
- 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 dayHeart 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.