GARMIN INTERNATIONAL

Garmin Developer Program Activity API

Version 1.2.2

CONFIDENTIAL

- **Contents**

1. Revision History

Version	Date	Revisions
1.0	12/01/2020	Initial version
1.0.2	08/02/2021	Backfill policy updated
1.0.3	11/25/2022	Appendix A updated
1.0.4	02/06/2023	Activity File JSON updated. Manual flag definition updated. Activity Details fields definition updated.
1.0.5	04/26/2023	Appendix A updated
1.0.6	30/08/2023	New wheelchair fields added to Activity summaries. Appendix A updated with new Activity Types
1.0.7	10/10/2023	Activity File JSON updated. Appendix A updated (typo corrected for RACQUETBALL and MOUNTAINEERING)
1.0.8	10/16/2023	Removing user access token reference in preparation for token retiring.
1.0.9	10/16/2023	Added 'isWebUpload' field. Appendix A updated.
1.0.10	11/01/2024	24-hour limitation update for Activity Details
1.1.0	12/1/2024	Backfill policy updated
1.1.1	02/27/2025	RUCKING activity added, Appendix A updated
1.1.2	04/28/2025	MOBILITY activity added, Appendix A updated
1.1.3	06/06/25	Backfill policy updated

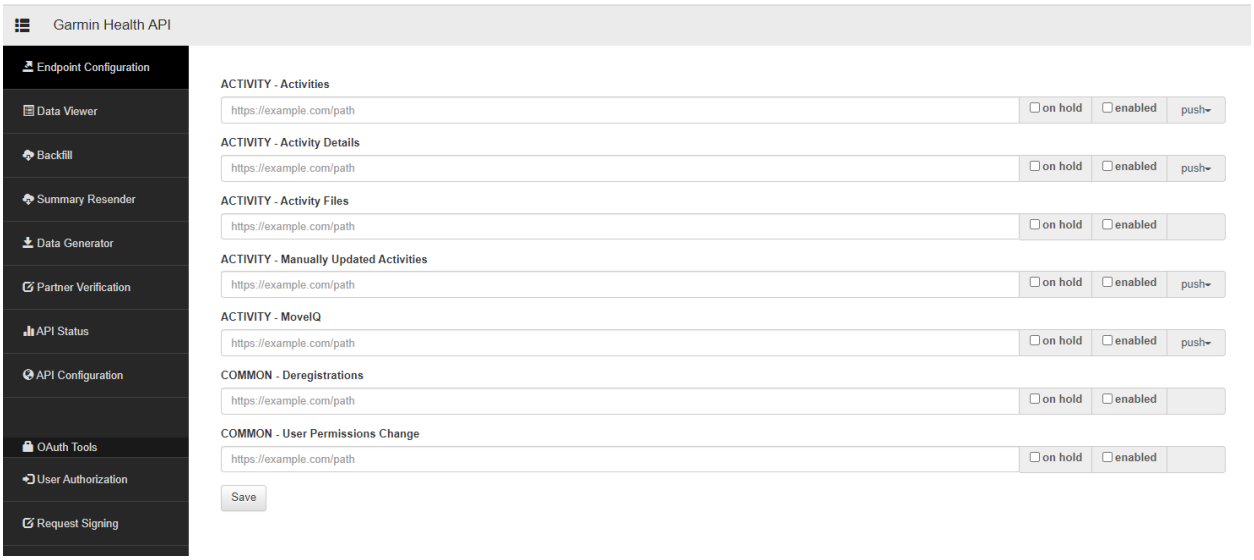
1.1.4	06/19/25	Enduro mountain biking activities are added
1.2.0	06/30/25	Production Requirements updated
1.2.1	07/21/2025	PADDELBALL activity name updated (PADDLE before)
1.2.2	08/07/2025	Activity File PING notification updated (‘devideName’ included)

2. Purpose of Activity API

The Activity API allows you to receive completed activity data captured on Garmin wearable devices and cycling computers. Fitness, training, wellness, or health tracking platforms can all benefit from leveraging the Activity API. After user consent, you can access the detailed fitness data logged by end-users

3. Endpoint Configuration

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



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

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

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

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

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

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

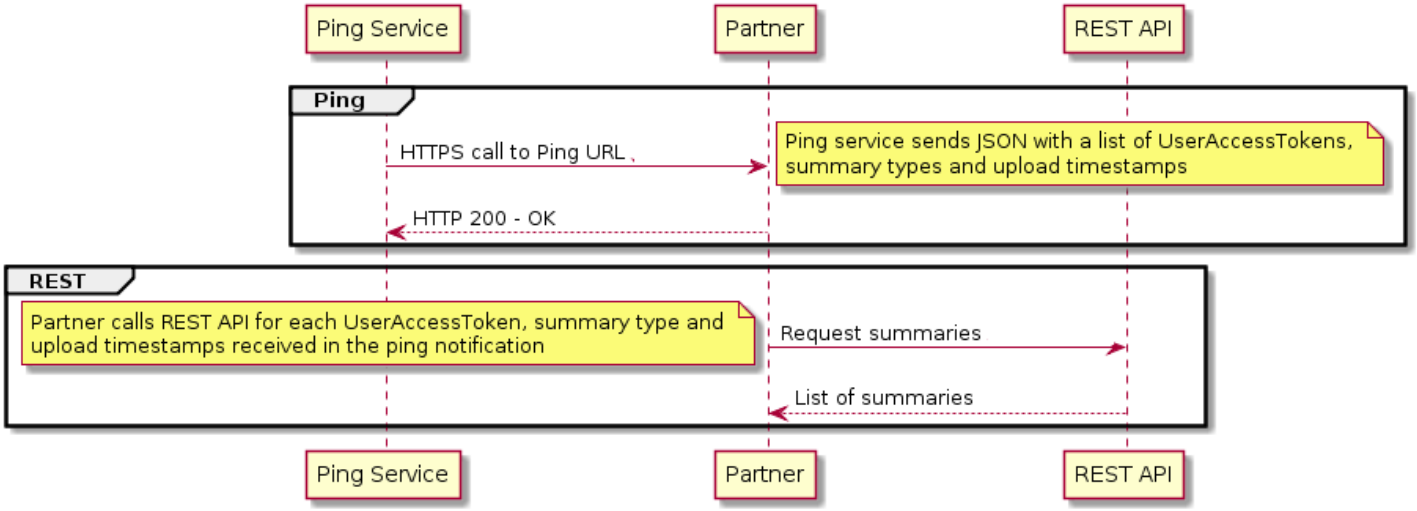
Each notification also contains a callback URL. When this URL is called, data specific to that user and summary type is returned. The partner may provide separate URLs for each summary type for flexible processing or may choose to send ping notifications for all data types to the same endpoint.

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

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

4.1 Ping Workflow

The following diagram illustrates the general workflow.



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

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

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

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

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

In the event of an unexpected outage in which notifications are accepted with HTTP 200s, but the resulting callbacks fail, please contact the Garmin Connect Developer Program Support team (connect-

support@developer.garmin.com). They will be happy to help set up a regeneration of all missed notifications during the affected time.

4.2 Ping Notification Content

JSON Element	Description
summary type (list key)	The summary type of this list of pings activities, activityDetails, activityFiles, moveIQActivities, manuallyUpdatedActivities
userId	A unique user identifier corresponding to the underlying Garmin account of the user. This userId is *not* used as a parameter for any call to the Activity API.
callbackURL	Pre-formed URL to pull the data. Not present for deregistration notifications.

Example

```
{
      "activities": [{
            "userId": "4aacafe82427c251df9c9592d0c06768",
            "callbackURL": "https://apis.garmin.com/wellness-
api/rest/activities?uploadStartTimeInSeconds=1444937651&uploadEndTimeI
nSeconds=1444937902"
      }]
}
```

Tip: During your Ping Service integration development, it may be cumbersome for your endpoints to be publicly available to receive real notifications from the Activity API. Simulating ping requests within the local network by using tools like cURL is a useful way to solve this problem.

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

```
curl -v -X POST -H "Content-Type: application/json;charset=utf-8" -d
'{"activities":[{"userId":
"4aacafe82427c251df9c9592d0c06768","uploadStartTimeInSeconds":14449376
51,"uploadEndTimeInSeconds":1444937902,"callbackURL":"https://apis.gar
min.com/wellness-
api/rest/activities?uploadStartTimeInSeconds=1444937651&uploadEndTimeI
nSeconds=1444937902"}]}' http://localhost:8080/garmin/ping
```

5. Push Service

Like the Ping Service, the Push Service allows partners to receive near-real-time updates of Garmin user data without delay or duplication associated with regularly scheduled update jobs. Unlike the Ping Service's callback URLs, the Push Service generates HTTPS POSTs that contain the updated data directly

within the POST as JSON. This data is the exact same data that would have been returned by the Activity API had a Ping notification been generated and its callback URL invoked; it is purely a matter of preference and ease of integration whether to use the Ping or Push Service.

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

5.1 Push Notification Content

JSON Element	Description
summary type (list key)	The summary type of this list of pings. activities, activityDetails, activityFiles, moveIQActivities, manuallyUpdatedActivities
userId	A unique user identifier corresponding to the underlying Garmin account of the user. This userId is *not* used as a parameter for any call to the Activity API. However, it will persist across userAccessTokens should the user re-register to generate a new UAT.
userAccessToken	The UAT corresponding to the user that generated the new data.
Summary data	The summary data in the same data model as the Activity API. See the Summary Endpoints section for details and examples of each summary data model.

Example


```
{
   "activities": [
      {
         "userId": "4aacafe82427c251df9c9592d0c06768",
         "summaryId": "EXAMPLE_12345",
           "activityType": "RUNNING",
           "startTimeInSeconds": 1452470400,
           "startTimeOffsetInSeconds": 0,
           "durationInSeconds": 11580,
           "averageSpeedInMetersPerSecond": 2.888999938964844,
           "distanceInMeters": 519818.125,
           "activeKilocalories": 448,
           "deviceName": "Forerunner 910XT",
           "averagePaceInMinutesPerKilometer": 0.5975272352046997
      }
   ]
}
```

6. Activity API Integration Tips

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

6.1 Time Values in the Activity API

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

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

6.2 Web Tools

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

6.2.1 Data Viewer

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

6.2.2 Backfill

The Backfill tool provides a web-based method to initiate historic data requests as described in the Summary Backfill section without the need to access the API programmatically.

6.2.3 Summary Resender

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

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

6.2.4 Data Generator

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

Please note that for the purposes of requesting a production-level key (see Requesting a Production Key above), data synced from actual devices is required.

6.2.5 Partner Verification

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

7. Summary Endpoints

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

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

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

7.1 Activity Summaries

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

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

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

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

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

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

Request

Resource URL

GET https://apis.garmin.com/wellness-api/rest/activities

Request parameters

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

Response

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

Each activity summary may contain the following parameters:

Property	Type	Description
summaryId	string	Unique identifier for the summary.
activityId	string	Unique identifier of the activity at Garmin Connect
startTimeInSeconds	integer	Start time of the activity in seconds since January 1, 1970, 00:00:00 UTC (Unix timestamp).
startTimeOffsetInSeconds	integer	Offset in seconds to add to startTimeInSeconds to derive the “local” time of the device that captured the data.
activityName	string	Garmin Connect activity name
activityType	string	Text description of the activity type. See Appendix A for a complete list.
durationInSeconds	integer	Length of the monitoring period in seconds.
pushes	integer	*This field will be present only if users’ watch is in wheelchair mode.
averageBikeCadenceInRoundsPerMinute	floating point	
averageHeartRateInBeatsPerMinute	integer	
averageRunCadenceInStepsPerMinute	floating point	
averagePushCadenceInPushesPerMinute	floating point	*This field will be present only if users’ watch is in wheelchair mode.
averageSpeedInMetersPerSecond	floating point	
averageSwimCadenceInStrokesPerMinute	floating point	

averagePaceInMinutesPerKilometer	floating point	
activeKilocalories	integer	Active kilocalories (dietary calories) burned during the monitoring period. This includes the calories burned by the activity and calories burned as part of the basal metabolic rate (BMR).
deviceName	string	Only Fitness Activities are associated with a specific Garmin device rather than the user's overall account. If a user wears two devices at once during the same time and starts a Fitness Activity on each then both will generate separate Activity summaries with two different deviceNames.
distanceInMeters	floating point	
maxBikeCadenceInRoundsPerMinute	floating point	
maxHeartRateInBeatsPerMinute	floating point	
maxPaceInMinutesPerKilometer	floating point	
maxRunCadenceInStepsPerMinute	floating point	
maxPushCadenceInPushesPerMinute	floating point	*This field will be present only if users' watch is in wheelchair mode.
maxSpeedInMetersPerSecond	floating point	
numberOfActiveLengths	integer	
startingLatitudeInDegree	floating point	
startingLongitudeInDegree	floating point	
steps	integer	
totalElevationGainInMeters	floating point	
totalElevationLossInMeters	floating point	
isParent	boolean	If present and set to true, this activity is the parent activity of one or more child activities that should also be made available in the data feed to the partner. An activity of type MULTI_SPORT is an example of a parent activity.
parentSummaryId	String	If present, this is the summaryId of the related parent activity. An activity of type CYCLING with a parent activity of type MULTI_SPORT is an example of this type of relationship.
manual	boolean	Indicates that the activity was generated not on Garmin Device, or manually created at Garmin Connect directly.
isWebUpload	boolean	Indicates that the activity was uploaded through the Garmin Connect Web app.

Example


```
[
{
        "summaryId" : "5001968355",
        "activityId" : 5001968355,
```

```
        “activityType”: “RUNNING”,
        “activityName”: “Olathe RUNNING”,
        “startTimeInSeconds”: 1452470400,
        “startTimeOffsetInSeconds”: 0,
        “durationInSeconds”: 11580,
        “averageSpeedInMetersPerSecond”: 2.888999938964844,
        “distanceInMeters”: 519818.125,
        “activeKilocalories”: 448,
        “deviceName”: “Garmi fenix 8”,
        “averagePaceInMinutesPerKilometer”: 0.5975272352046997
    },
    {
        "summaryId" : "5001968355",
        "activityId" : 5001968355,
        “activityType”: “CYCLING”,
        “activityName”: “Olathe CYCLING”,
        “startTimeInSeconds”: 1452506094,
        “startTimeOffsetInSeconds”: 0,
        “durationInSeconds”: 1824,
        “averageSpeedInMetersPerSecond”: 8.75,
        “distanceInMeters”: 4322.357,
        “activeKilocalories”: 360,
        “deviceName”: “Garmin fenix 8”
    }
]
```

7.2 Manually Updated Activity Summaries

Manual updated activities edited by the user directly on the Connect site and not uploaded from a device. Partners may choose to accept or ignore all or part of any manually created or updated Activities.

Request

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

Request parameters

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

Response

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

Each activity summary may contain the following parameters:

Property	Type	Description
summaryId	string	Unique identifier for the summary.
startTimeInSeconds	integer	Start time of the activity in seconds since January 1, 1970, 00:00:00 UTC (Unix timestamp).
startTimeOffsetInSeconds	integer	Offset in seconds to add to startTimeInSeconds to derive the “local” time of the device that captured the data.
activityType	string	Text description of the activity type. See Appendix A for a complete list.
durationInSeconds	integer	Length of the monitoring period in seconds.
averageBikeCadenceInRoundsPerMinute	floating point	
averageHeartRateInBeatsPerMinute	integer	
averageRunCadenceInStepsPerMinute	floating point	

averageSpeedInMetersPerSecond	floating point	
averagePushCadenceInPushesPerMinute	floating point	*This field will be present only if users' watch is in wheelchair mode.
averageSwimCadenceInStrokesPerMinute	floating point	
averagePaceInMinutesPerKilometer	floating point	
activeKilocalories	integer	
deviceName	string	Always 'unknown' for manually created activities.
pushes	integer	*This field will be present only if users' watch is in wheelchair mode.
distanceInMeters	floating point	
maxBikeCadenceInRoundsPerMinute	floating point	
maxHeartRateInBeatsPerMinute	floating point	
maxPaceInMinutesPerKilometer	floating point	
maxRunCadenceInStepsPerMinute	floating point	
maxPushCadenceInPushesPerMinute	Floating point	*This field will be present only if users' watch is in wheelchair mode.
maxSpeedInMetersPerSecond	floating point	
numberOfActiveLengths	integer	
startingLatitudeInDegree	floating point	
startingLongitudeInDegree	floating point	
totalElevationGainInMeters	floating point	
totalElevationLossInMeters	floating point	
isParent	boolean	If present and set to true, this activity is the parent activity of one or more child activities that should also be made available in the data feed to the partner. An activity of type MULTI_SPORT is an example of a parent activity.
parentSummaryId	integer	If present, this is the summaryId of the related parent activity. An activity of type CYCLING with a parent activity of type MULTI_SPORT is an example of this type of relationship.
Manual	boolean	Indicates that the activity was manually updated directly on the Connect site.

Example

Request:

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

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

Response:

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

]
```


7.3 Activity Details Summaries

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

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

Activity details summaries include all data recorded by the device as part of the Fitness Activity, including GPS coordinates and all recorded sensor data.

Please note that historical data is available only with PUSH Service.
The Activity Details summary endpoint enforces a duration limit of 24 hours. Any activity exceeding 24 hours in duration will not be transmitted or accessible through. Longer activities (over 24 hours) can be accessed via the Activity Files.

Request

Resource URL

GET https://apis.garmin.com/wellness-api/rest/activityDetails

Request parameters

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

Response

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

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

Property	Type	Description
summaryId	string	Unique identifier for the summary.
activityId	string	Unique identifier of the activity at Garmin Connect
startTimeInSeconds	integer	Start time of the activity in seconds since January 1, 1970, 00:00:00 UTC (Unix timestamp).
startTimeOffsetInSeconds	integer	Offset in seconds to add to startTimeInSeconds to derive the “local” time of the device that captured the data.
activityName	string	activityName
activityType	string	Text description of the activity type. See Appendix A for a complete list.
durationInSeconds	integer	Length of the monitoring period in seconds.
averageBikeCadenceInRoundsPerMinute	floating point	
averageHeartRateInBeatsPerMinute	integer	
averageRunCadenceInStepsPerMinute	floating point	
averagePushCadenceInPushesPerMinute	floating point	*This field will be present only if users’ watch is in wheelchair mode.
averageSpeedInMetersPerSecond	floating point	
averageSwimCadenceInStrokesPerMinute	floating point	
averagePaceInMinutesPerKilometer	floating point	
activeKilocalories	integer	
deviceName	string	Only Fitness activities are associated with a

		specific Garmin device rather than the user's overall account. If the user wears two devices at once at the same time and starts a Fitness Activity on each then both will generate separate Activities with two different deviceNames.
distanceInMeters	floating point	
maxBikeCadenceInRoundsPerMinute	floating point	
maxHeartRateInBeatsPerMinute	floating point	
maxPaceInMinutesPerKilometer	floating point	
maxRunCadenceInStepsPerMinute	floating point	
maxPushCadenceInPushesPerMinute	floating point	*This field will be present only if users' watch is in wheelchair mode.
maxSpeedInMetersPerSecond	floating point	
numberOfActiveLengths	integer	
startingLatitudeInDegree	floating point	
startingLongitudeInDegree	floating point	
steps	integer	
pushes	integer	*This field will be present only if users' watch is in wheelchair mode.
totalElevationGainInMeters	floating point	
totalElevationLossInMeters	floating point	
isParent	boolean	If present and set to true, this activity is the parent activity of one or more child activities that should also be made available in the data feed to the partner. An activity of type MULTI_SPORT is an example of a parent activity.
parentSummaryId	String	If present, this is the summaryId of the related parent activity. An activity of type CYCLING with a parent activity of type MULTI_SPORT is an example of this type of relationship.
manual	boolean	Indicates that the activity was generated not on Garmin Device, or manually created at Garmin Connect directly.

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

Property	Type	Description
startTimeInSeconds	integer	Start time of the sample in seconds since January 1, 1970, 00:00:00 UTC (Unix timestamp).
latitudeInDegree	floating point	Latitude in decimal degrees (DD)
longitudeInDegree	floating point	Longitude in decimal degrees (DD)
elevationInMeters	floating point	
airTemperatureCelcius	floating point	
heartRate	Integer	Heart rate in beats per minute
speedMetersPerSecond	floating point	(not supported for pool swimming activities)
stepsPerMinute	floating point	
totalDistanceInMeters	floating point	
timerDurationInSeconds	integer	The amount of "timer time" in an activity
clockDurationInSeconds	integer	The amount of real-world "clock time" from the

		start of an activity to the end
movingDurationInSeconds	integer	The amount of “timer time” during which the athlete was moving (above a threshold speed). (not supported for pool swimming activities)
powerInWatts	floating point	The amount of power expended in watts
bikeCadenceInRPM	floating point	Cycling cadence in revolutions per minute
directWheelchairCadence	floating point	Wheelchair cadence in pushes per minute *This field will be present only if users’ watch is in wheelchair mode.
swimCadenceInStrokesPerMinute	floating point	Swim cadence in strokes per minute (not supported for pool swimming activities)

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

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

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

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

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

Request:

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

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

Response:

```
[
   {
      "summaryId" : "5001968355-detail",
      "activityId" : 5001968355,
      “summary”: {
```

```
    "durationInSeconds": 1789,
    "startTimeInSeconds": 1512234126,
    "startTimeOffsetInSeconds": -25200,
    "activityType": "RUNNING",
    "activityName": "Olathe RUNNING",
    "averageHeartRateInBeatsPerMinute": 144,
    "averageRunCadenceInStepsPerMinute": 84.0,
    "averageSpeedInMetersPerSecond": 2.781,
    "averagePaceInMinutesPerKilometer": 15.521924,
    "activeKilocalories": 367,
    "deviceName": "Garmin fenix 8",
    "distanceInMeters": 4976.83,
    "maxHeartRateInBeatsPerMinute": 159,
    "maxPaceInMinutesPerKilometer": 10.396549,
    "maxRunCadenceInStepsPerMinute": 106.0,
    "maxSpeedInMetersPerSecond": 4.152,
    "startingLatitudeInDegree": 51.053232522681355,
    "startingLongitudeInDegree": -114.06880217604339,
    "steps": 5022,
    "totalElevationGainInMeters": 16.0,
    "totalElevationLossInMeters": 22.0
},
"samples": [{
      "startTimeInSeconds" : 1669313992,
      "latitudeInDegree" : 38.832325832918286,
      "longitudeInDegree" : -94.74890395067632,
      "elevationInMeters" : 314.0,
      "heartRate" : 108,
      "speedMetersPerSecond" : 1.3250000476837158,
      "totalDistanceInMeters" : 1903.4200439453125,
      "timerDurationInSeconds" : 1460,
      "clockDurationInSeconds" : 1460,
      "movingDurationInSeconds" : 1379
      },
      {
      "startTimeInSeconds" : 1669314001,
      "latitudeInDegree" : 38.832390792667866,
      "longitudeInDegree" : -94.74878308363259,
      "elevationInMeters" : 314.20001220703125,
      "heartRate" : 109,
      "speedMetersPerSecond" : 1.315999984741211,
      "totalDistanceInMeters" : 1916.18994140625,
      "timerDurationInSeconds" : 1469,
      "clockDurationInSeconds" : 1469,
      "movingDurationInSeconds" : 1388
      }
],
"laps": [
    {
       "startTimeInSeconds": 1512234126
    },
    {
```

```
            "startTimeInSeconds": 1512234915
        }
    ]
  }
]
```

7.4 Activity Files

Activity details are also available as raw FIT, TCX, or GPX files (based on device). These are the actual files recorded by the wearable as part of the Fitness Activity, including GPS coordinates, all recorded sensor data, and any product-specific data that may not be exposed as part of the parsed Activity Details

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

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

Unlike normal Summaries, Activity Files are not available as a Push integration. Files are only available in response to a Ping by calling the specified callbackURL.

Please note that Activity Files endpoint will provide only Garmin Original activities that were created by Garmin devices and manually uploaded activities as well. "Manual": true field is provided to indicate that the activity was manually updated/ created directly on the Connect site, or "Manual": false is provided to indicate that activity was originated at Garmin Device.

The Ping's body is JSON formatted as follows:

```
{"activityFiles" : [ {
    "userId" : "4aacafe82427c251df9c9592d0c06768"
    "summaryId" : "10010727180-file",
    "fileType" : "FIT",
    "callbackURL": "https://apis.garmin.com/wellness-
api/rest/activityFile?id=XXX&token=YYY",
    "activityType": "RUNNING",
    "deviceName": "Garmin Fenix 8",
    "startTimeInSeconds" : 1617717902,
    "activityId" : 5001904988,
    "activityName" : "Olathe WALKING",
    "manual" : false
  },
{
   "userId" : "a099ba88-6c85-43ec-8b58-63d286683cda",
    "summaryId" : "10010728581-file",
    "fileType" : "FIT",
    "callbackURL": "https://apis.garmin.com/wellness-
api/rest/activityFile?id=XXX&token=YYY",
    "activityType": "RUNNING",
    "deviceName": "Garmin Fenix 8",
    "startTimeInSeconds" : 1614619219,
    "activityId" : 5001905361,
```

```
    "activityName" : "Flanders, Oudenaarde Tour 1 - Wortegem-
Petegem",
    "activityDescription" : "First part of an easy two-stage ride
on the very light rolling hills to the north east of Oudenaarde.
We go back with Stage 2 towards Gent via the Schelde river bike
path.",
    "manual" : false
  } ]}
```

Unlike a normal Ping body, the file type (TCX, GPX, or FIT) is specified in the filetype field and the callback URL specifies the Activity File by an ID rather than by the upload time range.
*Note: activityId – id of user's activity at Garmin Connect.
activityDescription – will be generated for TACX activities only and any other activities that set default description.
** Note: Callback url will be available for download for 24 hours only and should be downloaded once. Duplicate downloads will be rejected with HTTP 410 status.
*** Note: callback url contains a token as a parameter (this is not a user access token)

7.5 Move IQ Summaries

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

Due to their automatically-detected nature, Move IQ events are not considered a fitness activity, do not contain the same details as activities, and cannot be edited by the user with Garmin Connect. These events should be considered a labeled-timespan on top of normal Daily or Epoch summary details, matching their representation within Garmin Connect.
For more feature-level information on Move IQ events, please see: https://support.garmin.com/en-US/?faq=zgFpy8MShkArqAxCug5wC6&productID=73207&searchQuery=move%20Iq&tab=topics. Move IQ activities are also known as Automatic Activity Detection in older devices or documentation.

Request

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

Request parameters

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

	Note: This parameter corresponds to the value given in a Ping notification.

Response

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

Each Move IQ event summary may contain the following parameters:

Property	Type	Description
summaryId	string	Unique identifier for the summary.
calendarDate	string	The calendar date this summary would be displayed on in Garmin Connect. The date format is 'yyyy-mm-dd'.
startTimeInSeconds	float	Start time of the summary in seconds since January 1, 1970, 00:00:00 UTC (Unix timestamp).
offsetInSeconds	integer	Offset in seconds to add to startTimeInSeconds to derive the "local" time of the device that captured the data.
durationInSeconds	integer	The duration of the measurement period in seconds.
activityType	string	The activity type that has been identified for this timespan.
activitySubType	string	The activity subtype that has been identified for this timespan.

Response:

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

8. Summary Backfill

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

A backfill request returns an empty response immediately, while the actual backfill process takes place asynchronously in the background. Once backfill is complete, a notification will be generated and sent as if data for that time period was newly synced. Both the Ping Service and the Push Service are supported by Summary Backfill. The maximum date range (inclusive) for a single backfill request is 30 days, but it is permissible to send multiple requests representing other 30-day periods to retrieve additional data.

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

Production level keys are rate-limited to 10,000 days of data requested per minute per key.

User rate limit – 1 months since the first user connection

* Note: Duplicate Backfill requests are rejected with HTTP 409 status (duplicate requests – requests for already requested time period)

Request

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

Resource URL for activity details (available only with PUSH Service)
GET https://apis.garmin.com/wellness-api/rest/backfill/activityDetails

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

Request parameters

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

Response

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

Example

Request:

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

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

9. Requesting a Production Key

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

Production Review:

To initiate the review, please get in touch with connect-support@developer.garmin.com

1. Technical Review:

You can use the Partner Verification tool to ensure that the following technical requirements are met (please provide a screenshot or completed verification):

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

2. UX and Brand Compliance Review:

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

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

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

3. Account Set up

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

Appendix A – Activity Types

Below is the list of valid activity types referenced in Garmin Connect fitness activity summaries and corresponding response through API.

ACTIVITY	NAME VIA API
RUNNING	**RUNNING**
INDOOR RUNNING	INDOOR_RUNNING
OBSTACLE COURSE RACING	OBSTACLE_RUN
STREET RUNNING	STREET_RUNNING
TRACK RUNNING	TRACK_RUNNING
TRAIL RUNNING	TRAIL_RUNNING
TREADMILL RUNNING	TREADMILL_RUNNING
ULTRA RUNNING	ULTRA_RUN
VIRTUAL RUNNING	VIRTUAL_RUN
CYCLING	**CYCLING**
BMX	BMX
CYCLOCROSS	CYCLOCROSS
DOWNHILL BIKING	DOWNHILL_BIKING
EBIKING	E_BIKE_FITNESS
EMOUNTAINBIKING	E_BIKE_MOUNTAIN
eENDURO MOUNTAIN BIKING	E_ENDURO_MTB
ENDURO MOUNTAIN BIKING	ENDURO_MTB
GRAVEL/UNPAVED CYCLING	GRAVEL_CYCLING
INDOOR CYCLING	INDOOR_CYCLING
MOUNTAIN BIKING	MOUNTAIN_BIKING
RECUMBENT CYCLING	RECUMBENT_CYCLING
ROAD CYCLING	ROAD_BIKING
TRACK CYCLING	TRACK_CYCLING
VIRTUAL CYCLING	VIRTUAL_RIDE
HANDCYCLING	HANDCYCLING
INDOOR_HANDCYCLING	INDOOR_HANDCYCLING
GYM & FITNESS EQUIPMENT	**FITNESS_EQUIPMENT**
BOULDERING	BOULDERING
ELLIPTICAL	ELLIPTICAL
CARDIO	INDOOR_CARDIO
HIIT	HIIT
INDOOR CLIMBING	INDOOR_CLIMBING
INDOOR ROWING	INDOOR_ROWING

MOBILITY	MOBILITY
PILATES	PILATES
STAIR STEPPER	STAIR_CLIMBING
STRENGTH TRAINING	STRENGTH_TRAINING
YOGA	YOGA
MEDITATION	MEDITATION
SWIMMING	**SWIMMING**
POOL SWIMMING	LAP_SWIMMING
OPEN WATER SWIMMING	OPEN_WATER_SWIMMING
WALKING/INDOOR WALKING	**WALKING**
CASUAL WALKING	CASUAL_WALKING
SPEED WALKING	SPEED_WALKING
HIKING	**HIKING**
RUCKING	RUCKING
WINTER SPORTS	**WINTER_SPORTS**
BACKCOUNTRY SNOWBOARDING	BACKCOUNTRY_SNOWBOARDING
BACKCOUNTRY SKIING	BACKCOUNTRY_SKIING
CROSS COUNTRY CLASSIC SKIING	CROSS_COUNTRY_SKIING_WS
RESORT SKIING	RESORT_SKIING
SNOWBOARDING	SNOWBOARDING_WS
RESORT SKIING/ SNOWBOARDING	RESORT_SKIING_SNOWBOARDING_WS
CROSS COUNTRY SKATE SKIING	SKATE_SKIING_WS
SKATING	SKATING_WS
SNOWSHOEING	SNOW_SHOE_WS
SNOWMOBILING	SNOWMOBILING_WS
WATER SPORTS	**WATER_SPORTS**
BOATING	BOATING_V2, BOATING
FISHING	FISHING_V2, FISHING
KAYAKING	KAYAKING_V2, KAYAKING
KITEBOARDING	KITEBOARDING_V2, KITEBOARDING
OFFSHORE GRINDING	OFFSHORE_GRINDING_V2, OFFSHORE_GRINDING
ONSHORE GRINDING	ONSHORE_GRINDING_V2, ONSHORE_GRINDING
PADDLING	PADDLING_V2, PADDLING
ROWING	ROWING_V2, ROWING
SAILING	SAILING_V2, SAILING
SNORKELING	SNORKELING

STAND UP PADDLEBOARDING	STAND_UP_PADDLEBOARDING_V2, STAND_UP_PADDLEBOARDING
SURFING	SURFING_V2, SURFING
WAKEBOARDING	WAKEBOARDING_V2, WAKEBOARDING
WATERSKIING	WATERSKIING
WHITEWATER	WHITEWATER_RAFTING_V2, WHITEWATER_RAFTING
WINDSURFING	WINDSURFING_V2, WINDSURFING
TRANSITION	**TRANSITION_V2**
BIKE TO RUN TRANSITION	BIKE_TO_RUN_TRANSITION_V2, BIKE_TO_RUN_TRANSITION
RUN TO BIKE TRANSITION	RUN_TO_BIKE_TRANSITION_V2, RUN_TO_BIKE_TRANSITION
SWIM TO BIKE TRANSITION	SWIM_TO_BIKE_TRANSITION_V2, SWIM_TO_BIKE_TRANSITION
TEAM SPORTS	**TEAM_SPORTS**
AMERICAN FOOTBALL	AMERICAN_FOOTBALL
BASEBALL	BASEBALL
BASKETBALL	BASKETBALL
CRICKET	CRICKET
FIELD HOCKEY	FIELD_HOCKEY
ICE HOCKEY	ICE_HOCKEY
LACROSSE	LACROSSE
RUGBY	RUGBY
SOCCER/FOOTBALL	SOCCER
SOFTBALL	SOFTBALL
ULTIMATE DISC	ULTIMATE_DISC
VOLLEYBALL	VOLLEYBALL
RACKET SPORTS	**RACKET_SPORTS**
BADMINTON	BADMINTON
PADEL	PADDELBALL
PICKLEBALL	PICKLEBALL
PLATFORM TENNIS	PLATFORM_TENNIS
RACQUETBALL	RACQUETBALL
SQUASH	SQUASH
TABLE TENNIS	TABLE_TENNIS
TENNIS	TENNIS, TENNIS_V2
OTHER	**OTHER**
BOXING	BOXING
BREATHWORK	BREATHWORK

DANCE	DANCE
DISC GOLF	DISC_GOLF
FLOOR CLIMBING	FLOOR_CLIMBING
GOLF	GOLF
INLINE SKATING	INLINE_SKATING
JUMP ROPE	JUMP_ROPE
MIXED MARTIAL ARTS	MIXED_MARTIAL_ARTS
MOUNTAINEERING	MOUNTAINEERING
ROCK CLIMBING	ROCK_CLIMBING
STOPWATCH	STOP_WATCH
PARA SPORTS	**PARA_SPORTS**
WHEELCHAIR PUSH RUN	WHEELCHAIR_PUSH_RUN
WHEELCHAIR PUSH WALK	WHEELCHAIR_PUSH_WALK

Appendix B – Error Responses

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

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

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

Example

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

Response:

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

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

}


HTTP/1.1 409
Date           Wed, 03 Feb 2016 12:15:17 GMT
Server         Apache
Content-Length 118
Content-Type   application/json;charset=utf-8

{
     errorMessage":"[6efb2a74-fa98-4d1c-aeb9-238b223fb304]duplicate
backfill processed at 2021-07-01T07:05:57Z [2021-04-02T07:05:57Z to
2021-07-01T07:05:57Z]

}
```