GARMIN INTERNATIONAL

Garmin Developer API Programs
Women's Health API

Version 1.0.4

CONFIDENTIAL

- ## Contents

1. Revision History

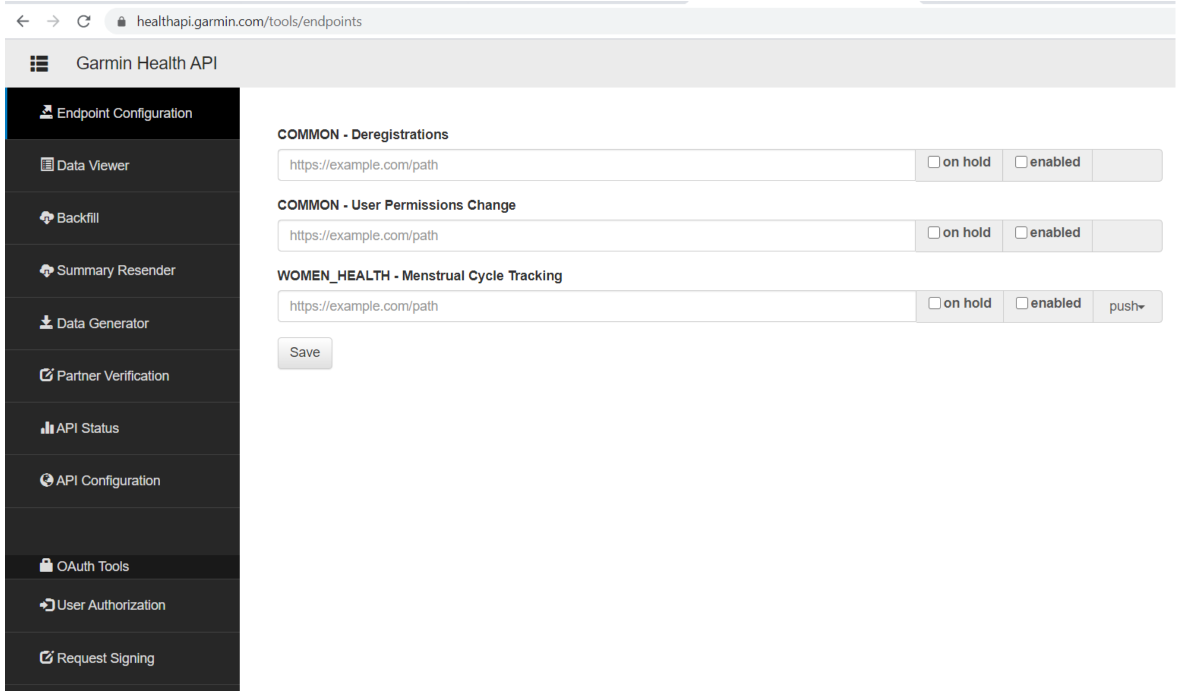
Version	Date	Revisions
1.0	12/01/2020	Initial revision
1.0.2	08/02/2021	Backfill policy updated for production level keys
1.0.3	10/16/2023	Removing reference for the user access token form PING/PUSH notifications examples in preparation to retire this field.
1.0.4	06/60/25	Backfill policy updated

2. Purpose of Women's Health API

The Garmin Connect Women's Health API lets you access information related to menstrual cycle tracking and schedule information. Similar to the Health API, these metrics are also applicable to the wellness population but additionally may be utilized alongside training plans and workout schedules. After user consent, you can access the data logged by end-users.

3. Endpoint Configuration

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



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

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

WOMEN_HEALTH - Menstrual Cycle Tracking
https://example.com/path □ on hold □ enabled push▾

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

Enabled: When checked, this summary data will be made available for all users associated with this consumer key and 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.

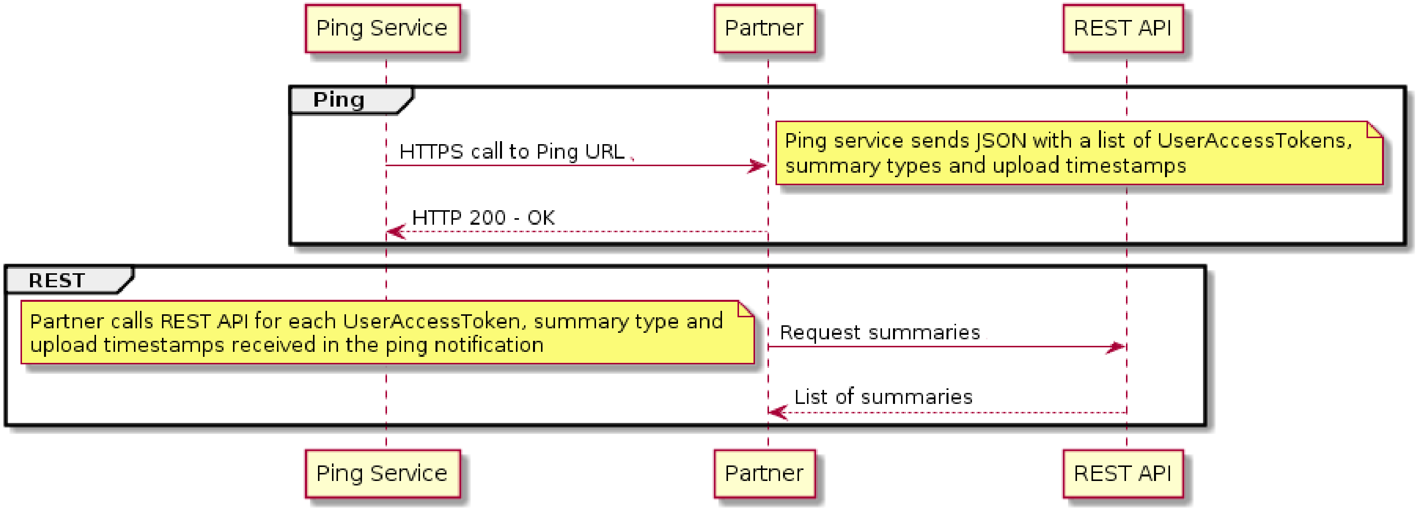
Each notification also contains a callback URL. When this URL is called, data specific to that user and summary type is returned. The partner may provide separate URLs for each summary type for flexible processing or may choose to send ping notifications for all data types to the same endpoint.

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

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 Women's Health API except as prompted by ping notifications.

4.1 Ping Workflow

The following diagram illustrates the general workflow.


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

The Ping Service has a timeout of thirty seconds. 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 Women's Health API. Holding the ping open while performing callbacks is the most common cause of instability in Women's Health API integrations.

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

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

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

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

In the event of an unexpected outage in which notifications are accepted with HTTP 200s, but the resulting callbacks fail, please contact the Garmin Connect Developer 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
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 Women’s Health API. However, it will persist across userAccessTokens should the user re-register to generate a new UAT.
userAccessToken	The UAT for which new data is available
uploadStartTimeInSeconds	The upload start timestamp of the new data in seconds since January 1, 1970, 00:00:00 UTC (Unix timestamp). Not present for deregistration notifications.
uploadEndTimeInSeconds	The upload end timestamp of the new data in seconds since January 1, 1970, 00:00:00 UTC (Unix timestamp). Not present for deregistration notifications.
callbackURL	Pre-formed URL to pull the data. Not present for deregistration notifications.

Example

```
{
      “mct”: [{
            “userId”: “4aacafe82427c251df9c9592d0c06768”,
            “userAccessToken”: “8f57a6f1-26ba-4b05-a7cd-c6b525a4c7a2”,
            “uploadStartTimeInSeconds”: 1444937651,
            “uploadEndTimeInSeconds”: 1444937902,
            “callbackURL”: “https://apis.garmin.com/wellness-
api/rest/mct?uploadStartTimeInSeconds=1444937651&uploadEndTimeInSecond
s=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 Women’s Health API. Simulating ping requests within the local network by using tools like cURL is a useful way to solve this problem.

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

```
curl -v -X POST -H “Content-Type: application/json;charset=utf-8” -d
‘{“epochs”:[{“userAccessToken”:”8f57a6f1-26ba-4b05-a7cd-
c6b525a4c7a2”,”uploadStartTimeInSeconds”:1444937651,”uploadEndTimeInSe
conds”:1444937902,”callbackURL”:”https://apis.garmin.com/wellness-
api/rest/mct?uploadStartTimeInSeconds=1444937651&uploadEndTimeInSecond
s=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 Women's Health API had a Ping notification been generated and its callback URL invoked; it is purely a matter of preference and ease of integration whether to use the Ping or Push Service.

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

5.1 Push Notification Content

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

Example

```
{
   “mct”: [
      {
         “userId”: “4aacafe82427c251df9c9592d0c06768”,
         “userAccessToken”: “8f57a6f1-26ba-4b05-a7cd-c6b525a4c7a2”,
         “summaryId”: “x2faf44e-16cfcc8d020”,
        “periodStartDate”: “2019-09-03”,
        “dayInCycle”: 1,
        “periodLength”: 1,
        “currentPhase”: 1,
        “lengthOfCurrentPhase”: 1,
        “daysUntilNextPhase”: 1,
        “predictedCycleLength”: 16,
        “isPredictedCycle”: false,
        “cycleLength”: 16,
        “lastUpdatedTimeInSeconds”: 1567609114
      }
   ]
}
```

6. Women's Health API Integration Tips

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

6.1 Time Values in the Women's Health API

All timestamps in the Women's Health API are UTC in seconds, also known as Unix Time. However, summary data records may also contain a time offset value. This value represents the difference between the standardized UTC timestamp.

6.2 Web Tools

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

6.2.1 Data Viewer

The Data Viewer tool allows viewing of a user's Women's Health API data by summary start and end time for the purposes of debugging or assisting an end user. This is the same data that can be pulled from the 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 API 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 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 Women's Health API, with each summary corresponding to a different ping notification type.

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

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

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

7.1 Menstrual Cycle Tracking (MCT) Summaries

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

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

Request

Resource URL

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

Request parameters

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

Response

A successful response is a JSON array containing zero to one MCT summaries. Each MCT summary contains menstrual cycle data and an optional pregnancy snapshot. The pregnancy snapshot will be empty when the user is not in a pregnant phase. Please see Appendix A for possible error responses.

Each MCT summary may contain the following parameters:

Property	Type	Description
summaryId	string	Unique identifier for the summary
periodStartDate	string	The calendar date representing period start date. The date format is ‘yyyy-mm-dd’
dayInCycle	integer	Represents nth day in cycle
periodLength	integer	Number of days indicating how long a period usually last
currentPhase	integer	Indicates the phase in this cycle like menstruation, fertile, etc.,
currentPhaseType	String	Indicates the phase in this cycle.
lengthOfCurrentPhase	integer	Represents the length of current phase in days
daysUntilNextPhase	integer	Number of days remaining to reach the next predicted phase
predictedCycleLength	integer	Number of days predicted to be the current cycle length
isPredictedCycle	boolean	A boolean to show if this summary is a predicted cycle or not
cycleLength	integer	A user logged cycle length

lastUpdatedTimeInSeconds	integer	Time in seconds showing when a user logged their symptoms
hasSpecifiedCycleLength	boolean	A boolean to show if cycle lenght was provided by the user through Garmin Connect
hasSpecifiedPeriodLength	boolean	A boolean to show if period length value was provided by the user through Garmin Connect

If a user is in a pregnant phase, the pregnancySnapshot field will be populated with the following fields:

Property	Type	Description
title	String	Title used for the user's pregnancy
originalDueDate	String	The calendar date representing the user's original due date entered by the user
dueDate	String	The calendar date representing an updated due date. This value is entered by the user in Garmin Connect.
pregnancyCycleStartDate	String	The calendar date representing the start of the user's pregnancy cycle
numOfBabies	String	A string representing the number of babies during the user's pregnancy
weightGoalUserInput	Dictionary	A representation of the user's height in centimeters (heightInCentimeters) and weight in grams (weightInGrams). This is entered by the user when the user is marked as pregnant in Garmin Connect. Please see below example for data structure.
bloodGlucoseList	Dictionary	A representation of the user-logged blood glucose readings. This contains valueInMilligramsPerDeciliter (floating point), logType (String), and reportTimestampInSeconds (integer). Please see below example for data structure.

Example

Request:

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

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

Response:

Example where user is marked as pregnant

```
[
        {
```

```
"summaryId": "x2faf44e-175f69ac4fa",
"periodStartDate": "2020-10-11",
"dayInCycle": 110,
"currentPhase": 6,
"currentPhaseType": "SECOND_TRIMESTER",
"lengthOfCurrentPhase": 98,
"daysUntilNextPhase": 80,
"predictedCycleLength": 280,
"isPredictedCycle": false,
"cycleLength": 280,
"lastUpdatedTimeInSeconds": 1606160139,
"hasSpecifiedCycleLength": false,
"hasSpecifiedPeriodLength": false,
"pregnancySnapshot": {
    "title": "Pregnancy 2021",
    "originalDueDate": "2021-07-17",
    "dueDate": "2021-07-17",
    "pregnancyCycleStartDate": "2020-10-11",
    "numOfBabies": "SINGLE",
    "weightGoalUserInput": {
        "heightInCentimeters": 162,
        "weightInGrams": 58966
    },
    "bloodGlucoseList": [
        {
            "valueInMilligramsPerDeciliter": 97.0,
            "logType": "BEFOREBED",
            "reportTimestampInSeconds": 1607032423
        },
        {
            "valueInMilligramsPerDeciliter": 92.0,
            "logType": "BEFOREMEAL",
            "reportTimestampInSeconds": 1607032403
        },
        {
            "valueInMilligramsPerDeciliter": 89.0,
            "logType": "AFTERMEAL",
            "reportTimestampInSeconds": 1607356249
        },
        {
            "valueInMilligramsPerDeciliter": 93.0,
            "logType": "AFTERMEAL",
            "reportTimestampInSeconds": 1607523916
        },
        {
            "valueInMilligramsPerDeciliter": 86.0,
            "logType": "OTHER",
            "reportTimestampInSeconds": 1607536280
        },
        {
            "valueInMilligramsPerDeciliter": 50.0,
            "logType": "BEFOREMEAL",
                    "reportTimestampInSeconds": 1611852498
                }
            ]
        }
    }
]
```

Example where user is in non-pregnant phase

```
[
    {
        "summaryId": "x153a9f3-176e4715000",
        "periodStartDate": "2021-01-04",
        "dayInCycle": 1,
        "periodLength": 5,
        "currentPhase": 1,
        "currentPhaseType": "MENSTRUAL",
        "lengthOfCurrentPhase": 5,
        "daysUntilNextPhase": 5,
        "fertileWindowStart": 11,
        "lengthOfFertileWindow": 7,
        "predictedCycleLength": 28,
        "isPredictedCycle": true,
        "cycleLength": 28,
        "lastUpdatedTimeInSeconds": 1610150400,
        "hasSpecifiedCycleLength": false,
        "hasSpecifiedPeriodLength": false,
        "pregnancySnapshot": {}
    }
]
```

```
[
    {
        "summaryId": "x153a9f3-176e4715000",
        "periodStartDate": "2021-01-04",
        "dayInCycle": 1,
        "periodLength": 5,
        "currentPhase": 1,
        "currentPhaseType": "MENSTRUAL",
        "lengthOfCurrentPhase": 5,
        "daysUntilNextPhase": 5,
        "fertileWindowStart": 11,
        "lengthOfFertileWindow": 7,
        "predictedCycleLength": 28,
        "isPredictedCycle": true,
        "cycleLength": 28,
        "lastUpdatedTimeInSeconds": 1610150400,
        "hasSpecifiedCycleLength": false,
        "hasSpecifiedPeriodLength": false,
        "pregnancySnapshot": {}
    }
]
```

8. Summary Backfill

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

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

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

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

User rate limit – 1 month since first Backfill request time stamp.

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

Request

Resource URL for mct summaries

GET https://ghapi-kc3.garmin.com/wellness-api/rest/backfill/mct

Request parameters

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

Response

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

Example

Request:

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

This request triggers the backfill of mct 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. To obtain a production-level key that is not rate-limited, your integration must be verified using the Partner Verification Tool.

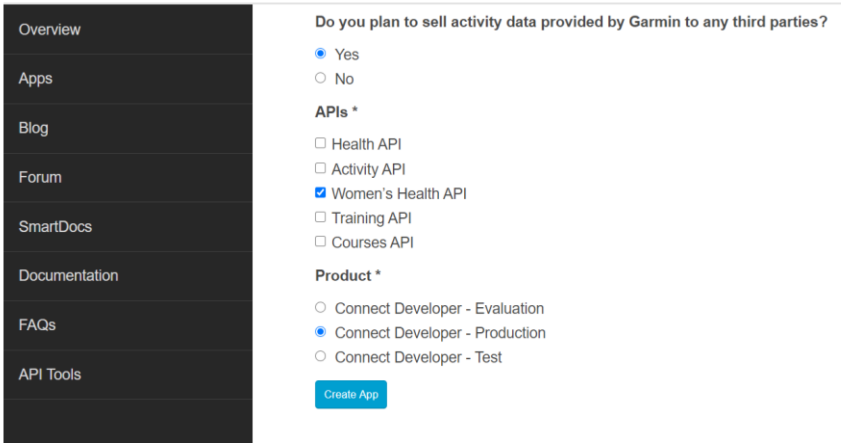
Tip: Before requesting a production key, please make sure your integration meets these basic requirements:

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

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

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

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



Do you plan to sell activity data provided by Garmin to any third parties?

● Yes
○ No

APIs *

□ Health API
□ Activity API
☑ Women's Health API
□ Training API
□ Courses API

Product *

○ Connect Developer - Evaluation
● Connect Developer - Production
○ Connect Developer - Test

Appendix A – 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.
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/mct?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."
}
```