Microsoft Health Cloud API

By using this document and the Microsoft Health Cloud API, you agree to be bound by the Terms of Use. Further, if accepting on behalf of a company, then you represent that you are authorized to act on your company's behalf.

CONTENTS

1 INTRODUCTION

Interested in building your new application or service to access Microsoft Health data? Want to integrate Microsoft Health data in your existing application? Now you can with the Microsoft Health Cloud API. The REST-based Microsoft Health Cloud APIs provides comprehensive user fitness and health data in an easy to consume format (JSON) that will allow you to enhance the experiences of your apps and services with real-time data from the Microsoft Health service.

2 GETTING STARTED

This document details the steps to get started with the Microsoft Health Cloud APIs.

2.1 ACCOUNT CREATION AND APPLICATION REGISTRATION

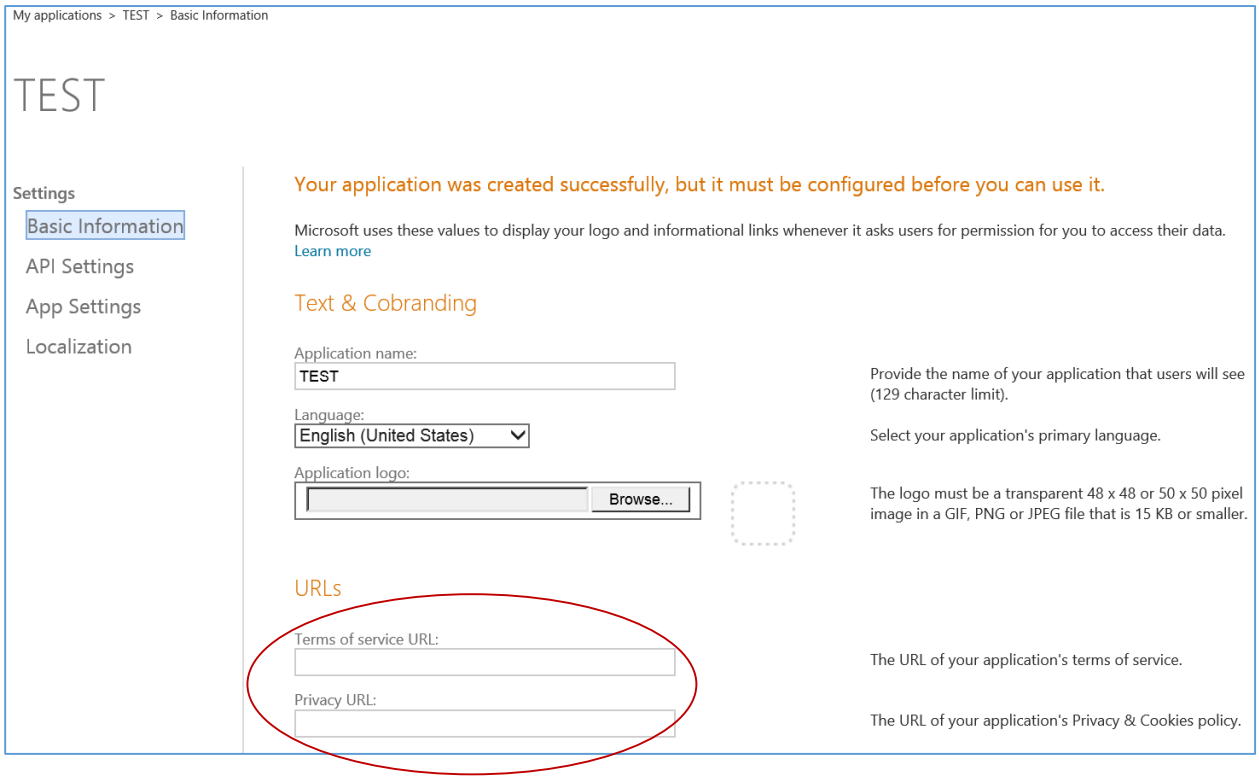
In order to connect to the Microsoft Health Cloud APIs, you will need a Microsoft Account with a registered application.

Remember that each app registered with Microsoft Account Developer Center is associated with the Microsoft account used for login to https://account.live.com/developers/applications. **We recommend that you use a developer account instead of a personal account.**

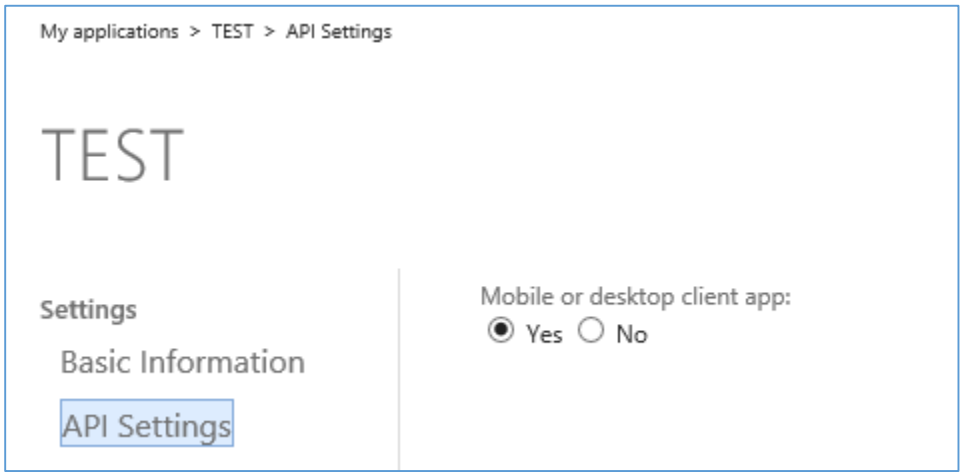
- To learn more about developer accounts, please visit https://msdn.microsoft.com/en-us/library/windows/apps/hh868184.aspx
- To sign up for a Microsoft account, please visit http://account.microsoft.com.
- Please make sure your Microsoft account is associated with a valid email address so we can keep you up-to-date on our latest status and releases.

To register your application in the Microsoft Account Developer Center, visit https://account.live.com/developers/applications. This will provide the client id and client secret that can be used within your application to authorize against Microsoft Health Cloud APIs.

Include URLs for your application's privacy policy and Terms of Use:



If developing an application, please select "Yes" for Mobile or Desktop Client app" in "My applications/<your app name>/API Settings".



Look in the [Application Authentication Scheme and Workflow] section or sample application for information about using client id and secret to build authorization workflows within your application.

2.2 Authentication Scheme

Microsoft Health Cloud API uses “Microsoft account (formerly Live Id)” token-based OAuth (Open Authorization) 2.0 authentication, a standard for token-based authentication and authorization.

OAuth allows an end user's account information to be used by third-party services, such as Microsoft Account, without exposing the user's password. To read more about OAUTH 2.0, please visit http://tools.ietf.org/html/rfc6749.

Every Microsoft Health Cloud API call requires an access token using an HTTPS header in the request, in the following format:

```
Authorization: Bearer {token}
```

Please refer to the Application Authentication Scheme and Workflow section below or our sample app for information about building authorization workflows using OAuth within your application or service.

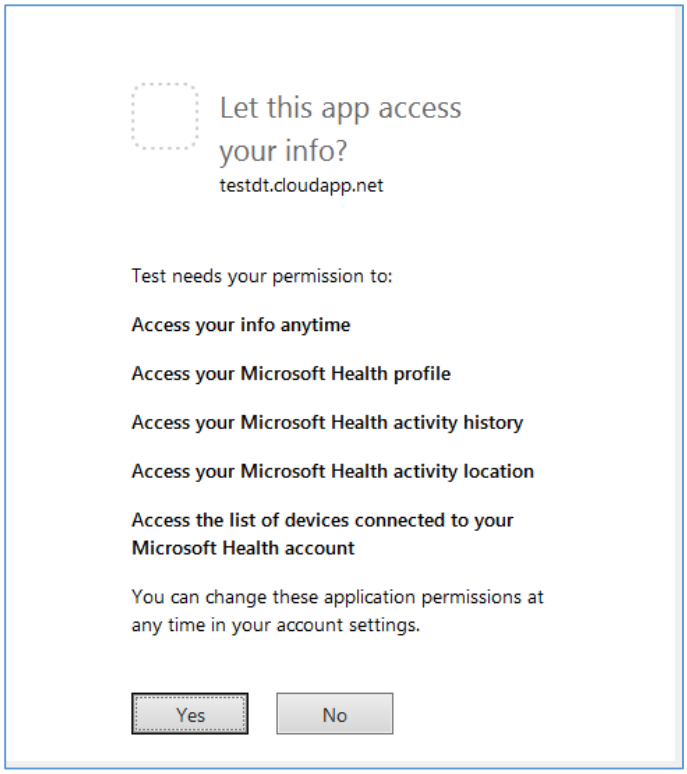
2.3 Authorization Scopes

The Microsoft Health Service supports five different scopes (or access types) for different types of user data. The developer needs to explicitly request authorization from each user to access their Microsoft Health data.

Scope Name	Scope Description
mshealth.ReadProfile	ReadProfile will be able to access your profile. Your profile includes things like your name, gender, weight, and age. Your email address will not be shared.
mshealth.ReadActivityHistory	ReadActivityHistory will be able to access your daily and historical activity information. Your activity history includes things like your runs, workouts, sleep, and daily steps.
mshealth.ReadDevices	ReadDevices will be able to access information about the devices associated with your Microsoft Health account.
mshealth.ReadActivityLocation	ReadActivityLocation will be able to access the location information for your activities.

offline_access	Offline_access will be able to receive a refresh token so it can work offline even when the user isn't active.

Based on scopes, the following will be shown to user at the consent page of the app or service.



Once the user authorizes access, Microsoft account will provide your application with a delegation ticket that you can use to authenticate with the Microsoft Health Cloud APIs. Please look in the Application Authentication Scheme and Workflow section below if you need more details about application authentication and the authorization workflow.

3 Using the API

The Microsoft Health Cloud API is a cloud-based service that enables developers to access health and fitness data via REST APIs. All requests are made over HTTPS and the data is presented using JSON.

3.1 Accessing the API

In order to make a successful call to Microsoft Health Cloud API, the following is required:

- A URL formed to meet the requirements of the desired request. The host for all API requests is api.microsofthealth.net

- Valid version
- The OAuth token issued to application session in the header

A valid example of such a request is given below

```
GET /v1/me/Profile HTTP/1.1
Authorization: bearer EwCoAvF0BAAUkWhN6f8bO0+=
```

If the API request is successful, the response will be the JSON with 200 return code.

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
{
  "firstName": "Jon",
  "lastName": "Dumont",
  "lastUpdateTime": "2015-05-20T18:39:29.461+00:00",
  "birthdate": "1980-03-01T20:44:18.851+00:00",
  "postalCode": "",
  "gender": "Male",
  "height": 1680,
  "weight": 5465,
  "preferredLocale": "en-us"
}
```

Please refer to the API Request Parameters section to see the parameters for every API.

3.2 API URLs

API	Verb	Description
V1/me/Profile	GET	Gets the user profile for the specified user.
V1/me /Devices	GET	Gets associated devices (Microsoft Band and phones) for a user.
V1/me /Devices/{DeviceId}	GET	Gets particular devices for a user.
V1/me /Activities	GET	Gets user activities for a specified date range or ID.
V1/me /Summaries/{period}	GET	Gets daily or hourly summaries for a date range. • Date ranges are start time plus two days for hourly summaries. • There is no range restriction for daily summaries.

The JSON response for each of the APIs is explained in Microsoft Health Objects section below.

3.3 VERSIONING

The Microsoft Health Cloud APIs are versioned so that existing applications and services are not affected when changes are introduced. The following are few example of the non-breaking versus breaking changes:

Non-Breaking changes	Breaking Changes
• Addition of new values in JSON response	• Removal of existing values in JSON response • Changing the type of values • Moving types into different sub-groups or entities

Versions are specified as part of the URL, required on every request. The following example illustrates how the parameter is specified:

```
GET https://api.microsofthealth.net/v1/me/Profile/
```

We recommend that you use the same version for a particular URL request across your application. This is especially true when new API versions introduce attributes or operations that are not recognized by previous versions. Mixing API version can have unintended consequences and should be avoided.

3.4 Time & Time Zones

The Microsoft Health Cloud APIs use the ISO 8601 time format and support both UTC and local time. Note that you must also URL encode the times. For example, request to get daily summaries looks like the following:

```
Local Time

/v1/me/Summaries/Daily?startTime=2015-05-05T16%3A04%3A49.8578590-07%3A00&endTime=2015-05-06T16%3A04%3A49.840-07%3A00


UTC Time

/v1/me/Summaries/Daily?startTime=2015-05-05T23%3A16%3A23.369Z&endTime=2015-05-06T23%3A16%3A23.3478076Z
```

Following ISO8601 convention, the time duration is specified in P[n]DT[n]H[n]M[n]S format.

3.5 Throttling & Usage Restrictions

The Microsoft Health service limits requests on a per-user basis, or more accurately, per access token. These limits are in place to make sure that individuals and apps do not adversely affect the experience of other users. The rate limit is given below.

Max Requests Per Second	Max Requests Per Minute	Response Bandwidth per minute	Timeout for violation
60	500	1 MB	30 seconds

When an application exceeds the rate limit for a given API endpoint, the Microsoft Health Cloud API will return an HTTP 429 “Too Many Requests” response code and API requests will be rejected for 30 seconds.

If you hit the rate limit on a given endpoint, this is the body of the HTTP 429 message that you will see:

```
{
  "error": {
    "code": " TooManyRequests ",
    "message": “Client application has been throttled and should not attempt to repeat the request until an amount of time has elapsed.”,
```

```
  }
}
```

We ask that you honor the rate limit. If your application abuses the rate limits, it will be blocked, and you will be unable to get a response from the API. If your application has been blocked and you think there has been an error, you can contact the email address in the Support section.

3.6 PAGINATION

The Microsoft Health Cloud API supports pagination on requests that can respond with large collections, such as the collection returned by activities or summaries APIs. Each of these responses contains a limited number of items set by `maxPagesize`.

The maximum page size for different API requests is as follows:

Summaries (Daily)	31
Summaries (Hourly)	48
Activities	1000

If the response is paged, it will also contain a nextPage property. To obtain the next page of items, pass this value of nextPage in the next API request. Repeat this process to page through the full collection. For the last page, nextPage will be absent.

For an example of pagination, calling the summaries method returns a response with nextPage:

```
{
  "summaries": [
    …
  ],
  "nextPage": "https://api.microsofthealth.net/v1/me/Summaries/Daily?s
tartTime=2015-04-20T16:07:22.2571848-07:00&endTime=2015-05-10T16:07:22
.2571848-07:00&MaxPageSize=5&ct=635664672000000000",
  "itemCount": 10
}
```

To get the next page of activities, simply pass the value of this URL in the next request:

```
GET https://api.microsofthealth.net/v1/me/Summaries/Daily?startTime=2015-04-20T16:07:22.2571848-07:00&endTime=2015-05-10T16:07:22.2571848-07:00&MaxPageSize=5&ct=635664672000000000
```

As before, the response to this request includes nextPage, which can be passed in to get the next page of results. Continue this cycle to get new pages.

Note: Page URLs become stale over time, since the Microsoft Health app is continually uploading data to the Microsoft Health Cloud. If new items have been added to a list since you started paginating, they might not appear in the results. If you hold a page URL for some time and want to continue paging, it might be better to restart pagination by repeating the original request.

3.7 Error Codes

Errors are returned using standard HTTP error code syntax. The following HTTP status codes should be expected.

Status code	Status message	Description
400	Bad Request	Cannot process the request because it is malformed or incorrect.
401	Unauthorized	Required authentication information is either missing or not valid for the resource.
403	Forbidden	Access is denied to the requested resource. The user might not have enough permission.
404	Not Found	The requested resource doesn't exist.
429	Too Many Requests	Client application has been throttled and should not attempt to repeat the request until an amount of time has elapsed.
500	Internal Server Error	There was an internal server error while processing the request.

The error response is a single JSON object. The object contains a single property named **error**, which includes all of the details of the error message. Additional information is included in the body of the failed call. Here is an example of a full JSON error body:

```
{
   "error": {
      "code": "NotFound",
      "message": "Device not found",
      "innererror": {
         "code": "NotFound",
         "message": "there was no device found for deviceid 2c7538fb-9cae-4987-9931-cd43604e641
9"
      }
   }
}
```

Important: Error messages are not localized and are intended for the developer to reference. They shouldn't be displayed directly to the user.

3.8 UNITS OF MEASUREMENTS

Microsoft Health Cloud API uses the following for units of measurement across the APIs, unless otherwise specified.

3.8.1.1 Metric System

Length	Centimeters (cm)
Weight	Grams (g)
Height	Millimeters (mm)
Temperature	Celsius (°C)
Time Duration	Timespan: P[n]DT[n]H[n]M[n]S
Speed	Centimeters / second (cm/s)
Pace	Milliseconds / meter (ms/m)

4 MICROSOFT HEALTH OBJECTS

Microsoft Health Cloud API introduces four different “objects” to the developer:

- User Profile
- Device
- Activity
- Summary

Consumers of these objects should tolerate the addition of new fields and variance in ordering of fields. Not all fields appear in all contexts. It is safe to consider a nulled field, an empty set, and the absence of a field as the same thing.

4.1 USERPROFILE

The UserProfile object contains the general profile of the person using Microsoft Band.

4.1.1.1 JSON Representation

```
{
  "firstName": "Jon",
  "middleName": null,
  "lastName": "Dumont",
  "lastUpdateTime": "2015-05-08T18:03:29.528+00:00",
  "birthdate": "1975-03-22T17:00:00",
  "postalCode": "98007",
  "gender": "Male",
  "height": 1750,
  "weight": 7000,
}
```

4.1.1.2 Properties

Property Name	***Type***	***Description***
FirstName	*String*	First name of the person
MiddleName	*String*	Middle name of the person
LastName	*String*	Last name of the person
LastUpdateTime	*DateTime*	The last time profile was updated

Birthdate	*DateTime*	Displays birth month and year of user (date is always the first of the month)
PostalCode	*String*	Postal or Zip code
Gender	*String*	Gender of the person
Height	*Integer*	Height of person
Weight	*Integer*	Weight of person

4.2 DEVICE

The Device object represents a device that collects and sends user data to the Microsoft Health service. An example of such a device is the Microsoft Band. Another example is the iPhone 5s or Microsoft Lumia 330, if the user has enabled the option to send steps to Microsoft Health App.

For more information on the multi-device scenario, please read
http://lumiaconversations.microsoft.com/2015/04/22/microsoft-health-and-microsoft-band-update/

4.2.1.1 JSON Representation

```
{
"deviceProfiles":
        {
        "id":" d3048b0c-f6a7-441b-967d-666d9ce5e986 ",
        "displayName":Microsoft Band,
        "lastSuccessfulSync":"2015-06-01T00:00:00+00:00"
         "deviceFamily":"Band"
        },
        {
        "id":"FFFF1300-FFFF-FFFF-6929-454E2D001160",
        "displayName":Nokia 630,
        "lastSuccessfulSync":"2015-01-01T00:00:00+00:00"
         "deviceFamily":"Windows"
}
```

4.2.1.2 Properties

Property Name	*Type*	*Description*
Id	String	The id of the device
DisplayName	String	Display name of the device
LastSuccessfulSync	DateTime	Last successful time of the device sync
DeviceFamily	Enum	Different device family types; supported values are Band, Windows, Android, iOS

4.3 ACTIVITY

The Activity object represents activities a user has completed using the tiles on the Microsoft Band. An example of an activity is sleep. Another example is a run activity.

The different activity types exposed by Microsoft Health are as follows:

- Run
- Bike
- Free Play (Workout)
- Guided Workout
- Golf
- Sleep

Each of these is discussed in detail in this section.

4.3.1 Run Activity

Run provides the relevant data for a user's running session. In addition to the general activity data, Run also provides the following:

- Performance summary
- Distance summary

4.3.1.1 JSON Representation

```
{
      "activityType": "Run",
      "pausedDuration": null,
      "splitDistance": 160934,
      "mapPoints": null,
      "id": "25197092999999999",
```

```
    "userId": null,
    "deviceId": null,
    "startTime": "2015-05-11T19:00:00+00:00",
    "endTime": "2015-05-11T21:00:00+00:00",
    "dayId": "2015-05-11T00:00:00+00:00",
    "createdTime": "0001-01-01T00:00:00+00:00",
    "createdBy": null,
    "name": "!^%@$^%$#^%!@$#^%@!$#^%@!",
    "duration": "PT2H",
    "performanceSummary": {
      "finishHeartRate": 55,
      "recoveryHeartRateAt1Minute": null,
      "recoveryHeartRateAt2Minutes": null,
      "heartRateZones": {
        "underAerobic": 120,
        "aerobic": null,
        "anaerobic": null,
        "fitnessZone": null,
        "healthyHeart": null,
        "redline": null,
        "overRedline": null
      }
    },
    "distanceSummary": {
      "period": "Activity",
      "totalDistance": 575920,
      "totalDistanceOnFoot": null,
      "actualDistance": 575920,
      "maxElevation": 0,
      "minElevation": 0,
      "waypointDistance": 2500,
```

```
        "speed": null,
        "pace": 1250173,
        "overallPace": null
      },
      "minuteSummaries": null,
      "caloriesBurnedSummary": {
        "period": "Activity",
        "totalCalories": 186
      },
      "heartRateSummary": {
        "period": "Activity",
        "averageHeartRate": 55,
        "peakHeartRate": 55,
        "lowestHeartRate": 55
      },
},
```

4.3.1.2 Properties

Property Name	***Type***	***Description***
PerformanceSummary	Object	Summary of performance data, see Performance Summary
DistanceSummary	Object	Summary of distance data, see Distance Summary
PausedDuration	Duration	Length of time the user was paused during the run
SplitDistance	Long	Split distance during the run
Id	*string*	The unique identifier of the activity
UserId	*string*	The user identifier of the person who completed the activity
DeviceId	*string*	The device (Microsoft Band) that created the activity
StartTime	*DateTime*	Start time of the activity
EndTime	*DateTime*	End time of the activity
DayId	*DateTime*	The date when this activity was created

CreatedTime	*DateTime*	The time when this activity was created
CreatedBy	*String*	The application that created this activity
Name	*String*	The name of this activity
Duration	*Duration*	The time length of the activity
ActivityType	*Object*	The activity type such as Run, Sleep, Bike, FreePlay (workout), Guided workout
Calories Burned Summary	*Object*	Summary of calories data; see CaloriesBurnedSummary
Heart Rate Summary	*Object*	Summary of heart rate data; see HeartRateSummary
Activity Segments(optional)	*Object*	Summary of activity segments; see Segments
Minute Summary (optional)	*Object*	Minute-by-minute summary during the activity; see Minute Details
Map points (optional)	*Object*	Summary of GPS points during the activity; see Map Points

4.3.2 Bike Activity

Bike provides the relevant data of a user's bike session. In addition to the general activity data, run also provides the following:

- Performance summary
- Distance summary

4.3.2.1 JSON Representation

```
"bikeActivities": [
    {
      "activityType": "Bike",
      "activitySegments": [

      ],
      "performanceSummary": {
        "finishHeartRate": 120,
        "recoveryHeartRateAt1Minute": 109,
        "recoveryHeartRateAt2Minutes": 102,
        "heartRateZones": {
```

```
        "overRedline": 2
      }
    },
    "distanceSummary": {
      "period": "Activity",
      "totalDistance": 11433,
      "actualDistance": 11492,
      "elevationGain": 200,
      "maxElevation": 11400,
      "minElevation": 11200,
      "waypointDistance": 2500
    },
    "splitDistance": 160934,
    "id": "2519726547949954190",
    "startTime": "2015-04-21T19:53:25.0045809+00:00",
    "endTime": "2015-04-21T19:54:56.0045809+00:00",
    "dayId": "2015-04-21T00:00:00+00:00",
    "createdTime": "0001-01-01T00:00:00+00:00",
    "name": "@!#$%^&*()_+{}|:\"<>?,.\/;[",
    "duration": "PT1M31S",
    "caloriesBurnedSummary": {
      "period": "Activity",
      "totalCalories": 13
    },
    "heartRateSummary": {
      "period": "Activity",
      "averageHeartRate": 120,
      "peakHeartRate": 125,
      "lowestHeartRate": 106
    }
```

```
    },
```

4.3.2.2 Properties

Property Name	***Type***	***Description***
PerformanceSummary	Object	Summary of performance data, see Performance Summary
DistanceSummary	Object	Summary of distance data, see Distance Summary
SplitDistance	Long	Split distance during the run
Id	*string*	The unique identifier of the activity
UserId	*string*	The user identifier of the person who completed the activity
DeviceId	*string*	The device (Microsoft Band) which created the activity
StartTime	*DateTime*	Start time of the activity
EndTime	*DateTime*	End time of the activity
DayId	*DateTime*	The date when this activity was created
CreatedTime	*DateTime*	The time when this activity was created
CreatedBy	*String*	The application which created this activity
Name	*String*	The name of this activity
Duration	*Duration*	The length of the activity
ActivityType	*Object*	The activity type such as Run, Sleep, Bike, FreePlay (workout), Guided workout
Calories Burned Summary	*Object*	Summary of calories related data, see CaloriesBurnedSummary
Heart Rate Summary	*Object*	Summary of heart rate data; see HeartRateSummary
Activity Segments(optional)	*Object*	Summary of activity segments; see Segments
Minute Summary (optional)	*Object*	Minute-by-minute summary during the activity; see Minute Details
Map points (optional)	*Object*	Summary of GPS points during the activity; see Map Points

4.3.3 Free Play (Workout) Activity

Free play (also called Workout in Microsoft Band Tiles) contains the relevant data from a user's workout activity.

In addition to the general activity data, run also providers the following:

- Performance summary
- Distance summary

4.3.3.1 JSON Representation


```
{
      "activityType": "FreePlay",
      "activitySegments": [

      ],
      "performanceSummary": {
        "finishHeartRate": 111,
        "recoveryHeartRateAt1Minute": null,
        "recoveryHeartRateAt2Minutes": null,
        "heartRateZones": {
          "underAerobic": null,
          "aerobic": null,
          "anaerobic": null,
          "fitnessZone": 60,
          "healthyHeart": null,
          "redline": null,
          "overRedline": null
        }
      },
      "distanceSummary": null,
      "pausedDuration": null,
      "splitDistance": null,
      "mapPoints": null,
      "id": "2519709371999999999",
      "userId": null,
      "deviceId": null,
      "startTime": "2015-05-11T17:00:00+00:00",
```

```
      "endTime": "2015-05-11T17:59:59+00:00",
      "dayId": "2015-05-11T00:00:00+00:00",
      "createdTime": "0001-01-01T00:00:00+00:00",
      "createdBy": null,
      "name": null,
      "duration": "PT59M59S",
      "minuteSummaries": null,
      "caloriesBurnedSummary": {
        "period": "Activity",
        "totalCalories": 413
      },
      "heartRateSummary": {
        "period": "Activity",
        "averageHeartRate": 111,
        "peakHeartRate": 114,
        "lowestHeartRate": 111
      },
      "properties": null
},
```


4.3.3.2 Properties

Property Name	***Type***	***Description***
PausedDuration	Duration	Length of the user was paused during the run
SplitDistance	Long	Split distance during the run
Id	*string*	The unique identifier of the activity
UserId	*string*	The user identifier of the person who completed the activity
DeviceId	*string*	The device (Microsoft Band) which created the activity
StartTime	*DateTime*	Start time of the activity
EndTime	*DateTime*	End time of the activity
DayId	*DateTime*	The date when this activity was created

CreatedTime	*DateTime*	The time when this activity was created
CreatedBy	*String*	The application which created this activity
Name	*String*	The name of this activity
Duration	*Duration*	The length of the activity
ActivityType	*Object*	The activity type such as Run, Sleep, Bike, FreePlay (workout), Guided workout
Calories Burned Summary	*Object*	Summary of calories related data, see CaloriesBurnedSummary
Heart Rate Summary	*Object*	Summary of heart rate data, see HeartRateSummary
Activity Segments(optional)	*Object*	Summary of activity segments, see Segments
Minute Summary (optional)	*Object*	Minute-by-minute summary during the activity; see Minute Details
Map points (optional)	*Object*	Summary of GPS points during the activity, see Map Points

4.3.4 Guided Workout Activity

Guided Workout provides the relevant data for a user's guided workout session. In addition to the general activity data, guided workout also provides the following:

- Performance summary

4.3.4.1 JSON Representation

```
"guidedWorkoutActivities": [
    {
      "activityType": "GuidedWorkout",
      "cyclesPerformed": 1,
      "roundsPerformed": 5,
      "repetitionsPerformed": 289,
      "workoutPlanId": "ST-MW-9016",
      "performanceSummary": {
        "finishHeartRate": 152,
        "recoveryHeartRateAt1Minute": 123,
        "recoveryHeartRateAt2Minutes": 110,
```

```
        "heartRateZones": {
          "underAerobic": 1,
          "aerobic": 5,
          "anaerobic": 4,
          "fitnessZone": 6,
          "healthyHeart": 1,
          "redline": 1
        }
      },
      "id": "2519709421219467015",
      "startTime": "2015-05-11T15:37:58.0532984+00:00",
      "endTime": "2015-05-11T15:53:58.0532984+00:00",
      "dayId": "2015-05-11T00:00:00+00:00",
      "createdTime": "0001-01-01T00:00:00+00:00",
      "name": "1st GWO",
      "duration": "PT16M",
      "caloriesBurnedSummary": {
        "period": "Activity",
        "totalCalories": 184
      },
      "heartRateSummary": {
        "period": "Activity",
        "averageHeartRate": 142,
        "peakHeartRate": 167,
        "lowestHeartRate": 76
      }
    }
  ],
```

4.3.4.2 Properties

Property Name	***Type***	***Description***
Id	*string*	The unique identifier of the activity
UserId	*string*	The user identifier of the person who completed the activity
DeviceId	*string*	The device (Microsoft Band) which created the activity
StartTime	*DateTime*	Start time of the activity
EndTime	*DateTime*	End time of the activity
DayId	*DateTime*	The date when this activity was created
CreatedTime	*DateTime*	The time when this activity was created
CreatedBy	*String*	The application which created this activity
Name	*String*	The name of this activity
Duration	*Duration*	The length of the activity
ActivityType	*Object*	The activity type such as Run, Sleep, Bike, FreePlay (workout), Guided workout
Calories Burned Summary	*Object*	Summary of calories related data, see CaloriesBurnedSummary
Heart Rate Summary	*Object*	Summary of heart rate data, see HeartRateSummary
Activity Segments(optional)	*Object*	Summary of activity segments, see Segments
Minute Summary (optional)	*Object*	Minute-by-minute summary during the activity, see Minute Details
Map points (optional)	*Object*	Summary of GPS points during the activity, see Map Points
CyclesPerfomed	*Int*	Number of complete workout cycles performed
RoundsPerformed	*Int*	Number of rounds/circuit rounds performed
RepetitionsPerformed Int	*Int*	Numbers of repetitions performed
WorkoutPlanId	*Int*	Workout plan id

4.3.5 Golf Activity

Golf provides the relevant data for a user's golf session.

4.3.5.1 JSON Representation

```
{
```

```
"golfActivities": [
  {
    "activityType": "Golf",
    "activitySegments": [
      {
        "holeNumber": 1,
        "stepCount": 32,
        "distanceWalked": 2496,
        "segmentId": 63569091608604,
        "startTime": "2015-06-05T09:00:08.603+00:00",
        "endTime": "2015-06-05T09:12:56.603+00:00",
        "duration": "PT12M48S",
        "heartRateSummary": {
          "period": "Unknown"
        },
        "caloriesBurnedSummary": {
          "period": "Unknown",
          "totalCalories": 18
        },
        "segmentType": "GolfHole"
      },
    "totalStepCount": 32,
    "totalDistanceWalked": 216,
    "parOrBetterCount": 2,
    "longestDriveDistance": 181790,
    "longestStrokeDistance": 182158,
    "id": "2519688059922051338",
    "startTime": "2015-06-05T09:00:07.794+00:00",
    "endTime": "2015-06-05T15:40:32.794+00:00",
    "dayId": "2015-06-05T00:00:00.000+00:00",
    "duration": "PT6H40M25S",
```

```
      "caloriesBurnedSummary": {
        "period": "Activity",
        "totalCalories": 522
      },
      "heartRateSummary": {
        "period": "Activity"
      }
    }
  ],
}
```


4.3.5.2 Properties

Property Name	***Type***	***Description***
Id	*string*	The unique identifier of the activity
UserId	*string*	The user identifier of the person who completed the activity
StartTime	*DateTime*	Start time of the activity
EndTime	*DateTime*	End time of the activity
DayId	*DateTime*	The date when this activity was created
ActivityType	*Object*	The activity type such as Run, Sleep, Bike, FreePlay (workout), Guided workout
TotalStepCount	*Int*	The total number of steps a user took during the activity
TotalDistanceWalked	*Int*	The total distance a user walked during the activity
ParOrBetterCount	*Int*	The number of holes played where the user scored par or better during the activity
LongestDriveDistance	*Int*	The distance of the longest drive hit by the user during the activity
LongestStrokeDistance	*Int*	The distance of the longest stroke hit by the user during the activity
Calories Burned Summary	*Object*	Summary of calories related data, see CaloriesBurnedSummary
Heart Rate Summary	*Object*	Summary of heart rate data, see HeartRateSummary

Activity Segments(optional)	*Object*	Summary of activity segments, see Golf Segments
Minute Summary (optional)	*Object*	Minute-by-minute summary during the activity; see Minute Details

4.3.5.3 Golf Segment

HoleNumber	*Int*	The hole number on the golf course
StepCount	*Int*	The steps taken by the user for the hole
DistanceWalked	*Int*	The distance walked by the user for the hole

4.3.6 Sleep Activity

Sleep provides the relevant data from a user's sleep session, regardless of whether the session was manually started or was auto-detected.

4.3.6.1 JSON Representation


```
{
      "activityType": "Sleep",
      "activitySegments": [],
      "awakeDuration": "P1DT3H23M",
      "sleepDuration": "P14DT9H42M",
      "numberOfWakeups": 3,
      "fallAsleepDuration": "PT8H57M",
      "sleepEfficiencyPercentage": 95,
      "totalRestlessSleepDuration": "PT4H35M30S",
      "totalRestfulSleepDuration": "PT1H10M12S",
      "restingHeartRate": 63,
      "fallAsleepTime": "2015-05-11T08:25:56.202+00:00",
      "wakeupTime": "2015-05-11T14:27:35.755+00:00",
      "id": "2519709685807976512",
      "userId": null,
      "deviceId": null,
      "startTime": "2015-05-11T08:16:59.2023487+00:00",
      "endTime": "2015-05-11T14:30:04.2023487+00:00",
      "dayId": "2015-05-11T00:00:00+00:00",
      "createdTime": "0001-01-01T00:00:00+00:00",
      "createdBy": null,
      "name": null,
      "duration": "PT6H13M5S",
      "minuteSummaries": null,
      "caloriesBurnedSummary": {
        "period": "Activity",
```

```
        "totalCalories": 492
      },
      "heartRateSummary": {
        "period": "Activity",
        "averageHeartRate": 74,
        "peakHeartRate": 110,
        "lowestHeartRate": 56
      },
      "properties": null
},
```


4.3.6.2 Properties

Property Name	***Type***	***Description***
Id	*string*	The unique identifier of the activity
UserId	*string*	The user identifier of the person who completed the activity
DeviceId	*string*	The device (Microsoft Band) which created the activity
StartTime	*DateTime*	Start time of the activity
EndTime	*DateTime*	End time of the activity
DayId	*DateTime*	The date when this activity was created
CreatedTime	*DateTime*	The time when this activity was created
CreatedBy	*String*	The application which created this activity
Name	*String*	The name of this activity
Duration	*Duration*	The length of the activity
ActivityType	*Object*	The activity type such as Run, Sleep, Bike, FreePlay (workout), Guided workout
Calories Burned Summary	*Object*	Summary of calories related data, see CaloriesBurnedSummary
Heart Rate Summary	*Object*	Summary of heart rate data, see HeartRateSummary
Activity Segments(optional)	*Object*	Summary of activity segments, see Segments
Minute Summary (optional)	*Object*	Minute-by-minute summary during the activity, see Minute Details

Map points (optional)	*Object*	Summary of GPS points during the activity, see Map Points
AwakeDuration	Duration	Length of time user was awake during sleep session
SleepDuration	Duration	Total time of sleep activity
NumberOfWakeups	Int	Number of wakeups
FallAsleepDuration	Duration	Length of time it took user to fall asleep
SleepEfficiencyPercentage	Int	Ratio of time asleep to total sleep
TotalRestlessSleepDuration	Duration	Total length of restless sleep in minutes
TotalRestfulSleepDuration	Duration	Total length of restful sleep in minutes
RestingHeartRate	Int	Resting heart rate during the sleep
FallAsleepTime	DateTime	Date and time of the day user fell asleep
WakeupTime	DateTime	Date and time of the day user woke up

4.4 General Activity Data

This section captures the different activity data exposed by activities.

4.4.1 Heartrate Zones

Heart rate zones reflect exercise intensity and its effects on cardiovascular system.

4.4.1.1 JSON Representation

```
"heartRateZones": {
          "underAerobic": 1,
          "aerobic": 5,
          "anaerobic": 4,
          "fitnessZone": 6,
          "healthyHeart": 1,
          "redline": 1
```

4.4.1.2 Properties

Property Name	***Type***	***Description***
UnderAerobic	Int	Minutes under the Aerobic zone
Aerobic	Int	Minutes in the Aerobic zone
Anaerobic	Int	Minutes in the Anaerobic zone
FitnessZone	Int	Minutes in the fitness zone
Healthy Heart	Int	Minutes in the healthy heart zone
RedLine	Int	Minutes in the Redline zone
OverRedLine	Int	Minutes over the Redline zone

4.4.2 Activity Segment

Each of the activities in Microsoft Health can be divided into segments. These segments are created automatically when the user achieves a milestone (for example, reaching a mile mark during a run) or can be set by users (suspending and resuming an activity).

Each of the segments contains the same information as the activity, but the data is limited to that segment.

4.4.2.1 JSON Representation


```
"activitySegments": [
        {
          "segmentId": 635670725398665693,
          "startTime": "2015-04-22T01:04:12.133+00:00",
          "endTime": "2015-04-22T03:03:27.133+00:00",
          "duration": "PT1H59M15S",
          "heartRateSummary": {
            "period": " Activity",
            "averageHeartRate": 99,
            "peakHeartRate": 124,
            "lowestHeartRate": 70
          },
          "caloriesBurnedSummary": {
            "period": "Activity",
            "totalCalories": 517
          },
          "segmentType": "Run"
        }
      ],
```

4.4.3 Minute Details

The Minute summaries provide the details of every minute during the activity. It contains the same information as the activity, but is limited to that minute.

4.4.3.1 JSON Representation

```
"minuteSummaries": [
        {
          "userId": "89809f49-ba78-4d6e-9819-679b3185abf7",
          "startTime": "2015-04-16T20:05:00+00:00",
          "endTime": "2015-04-16T20:06:00+00:00",
          "period": "Minute",
          "duration": "PT1M",
          "stepsTaken": 0,
          "caloriesBurnedSummary": {
            "period": "Minute"
          },
          "heartRateSummary": {
            "period": "Minute"
          },
          "distanceSummary": {
            "period": "Minute",
            "totalDistance": 0,
            "totalDistanceOnFoot": 0
          }
        }
      ],
```

4.4.3.2 Properties

Property Name	***Type***	***Description***

UserId	String	Unique identifier of the user in the Microsoft Health service
StartTime	DateTime	Start time of the period
EndTime	DateTime	End time of the period
Period	Object	Length of time bucket; see Period
Duration	TimeSpan	Time duration in ISO8601 format
Steps	Int	Total number of steps taken during the duration
CaloriesBurnedSummary	Object	Summary of calories related data, see CaloriesBurnedSummary
HeartRateSummary	Object	Summary of heart rate data, see HeartRateSummary
DistanceSummary	Object	Summary of distance data, see DistanceSummary

4.4.4 Map Point

Map points help provide the location of the user at a particular time, from the starting point until the end of the activity. Requesting location information requires that the application or service requests explicit user consent to mshealth.ReadActivityLocation offer. For more on offers, read through the Authorization Scopes section.

4.4.4.1 JSON Representation

```
"mapPoints": [
        {
          "mapPointType": "Start",
          "ordinal": 0,
          "heartRate": 77,
          "scaledPace": 84,
          "isPaused": false,
          "isResume": false
        },
        {
          "secondsSinceStart":314,
          "mapPointType":"Waypoint",
          "ordinal":1,
          "actualDistance":2500,
          "totalDistance":2500,
          "heartRate":107,
          "pace":3030000,
          "scaledPace":2,
          "speed":33,
          "location":
          {
              "latitude":10,
              "longitude":20,
              "elevationFromMeanSeaLevel":30
               },
```

```
                "isPaused":false,
                "isResume":false
            },
          {
            "secondsSinceStart": 398,
            "mapPointType": "End",
            "ordinal": 25,
            "actualDistance": 6241,
            "totalDistance": 6240,
            "heartRate": 84,
            "scaledPace": 100,
            "isPaused": true,
            "isResume": false
          }
]
```

4.4.4.2 Properties

Property Name	***Type***	***Description***
SecondsSinceStart	Int	The number of seconds that have elapsed since mapping began
MapPointType	Object	Start, End, Split, Waypoint
Ordinal		Absolute ordering of this point relative to other points
ActualDistance	Long	Distance, not including distance travel while activity was paused
TotalDistance	Long	Distance from start point to this map point
HeartRate	Int	Heart rate at the closest time to this map point during the activity
Pace	Int	Time divided by Total Distance
ScaledPace	Int	An integer between 0 and 100 which represents the place between slowest and fastest pace; used to generate heat map or graph.
Speed	Int	Actual Distance divided by time

Location	Object	An object of latitude and longitude; see GPS Point
IsPaused	Bool	Whether or not this map point occurred during paused time
IsResume	Bool	Whether or not this map point is the first one since the activity was resumed

4.4.5 GPS Point

GPS point provides the latitude and longitude related data for an activity. This value is set only when GPS is enabled.

4.4.5.1 Properties

Property Name	***Type***	***Description***
SpeedOverGround	Int	Speed over ground in meters/second
Latitude	Int	Latitude in degrees
Longtitude	Int	Longitude in degrees
ElevationFromMeanSeaLevel	Int	Elevation from mean sea level
EstimatedHorizontalError	Int	Estimated horizontal error in meters
EstimatedVerticalError	Int	Estimated vertical error in meters

4.5 SUMMARY

The Summary object provides a sum-up of user data on an hourly or daily basis. This data is divided into several sub-groups:

Steps	The total number of steps taken in the time period
Calories Burned	The total calories burned in the time period
Heart Rate	The average, peak, and lowest heart rate in the time period
Distance	Distance-related measures in the time period

4.5.1.1 JSON Representation


```
{
  "summaries": [
    {
      "userId": "1560ce1d-0cfb-4ced-9766-4dcfb2166034",
      "startTime": "2015-05-10T00:00:00+00:00",
      "endTime": "2015-05-11T00:00:00+00:00",
      "period": "Daily",
      "duration": "P1D",
      "stepsTaken": 5877,
      "caloriesBurnedSummary": {
        "period": "Daily",
        "totalCalories": 2810
      },
      "heartRateSummary": {
        "period": "Daily",
        "averageHeartRate": 58,
        "peakHeartRate": 109,
        "lowestHeartRate": 45
      },
      "distanceSummary": {
        "period": "Daily",
```

```
        "totalDistance": 656384,
        "totalDistanceOnFoot": 656384,
        "actualDistance": null,
        "maxElevation": null,
        "minElevation": null,
        "waypointDistance": null,
        "speed": null,
        "pace": null,
        "overallPace": null
      },
      "activeHours": null
  ],
}
```


4.5.1.2 Properties

Property Name	***Type***	***Description***
UserId	String	Unique identifier of the user in the Microsoft Health service
StartTime	DateTime	Start time of the period
EndTime	DateTime	End time of the period
Period	Object	Length of time bucket; see Period
Duration	TimeSpan	Time duration in ISO8601 format
Steps	Int	Total number of steps taken during the duration
CaloriesBurnedSummary	Object	Summary of calories related data, see CaloriesBurnedSummary
HeartRateSummary	Object	Summary of heart rate data, see HeartRateSummary
DistanceSummary	Object	Summary of distance data, see DistanceSummary
ActiveHours	int	Number of active hours in the period; used for "daily summary"

4.5.1 Calories Burned Summary

Calories Burned provides calories burned during the period, using Microsoft Health algorithms.

Property Name	***Type***	***Description***
Period	Object	Length of time bucket; see Period
TotalCalories	Int	Total calories burned in the period

4.5.2 Heartrate Summary

Heartrate provides the heartrate data during the period, using Microsoft Health algorithms.

Property Name	***Type***	***Description***
Period	Object	Length of time bucket; see Period
AverageHeartRate	Byte	Average heart rate during the period
PeakHeartRate	Byte	Peak heart rate during the period
LowestHeartRate	Byte	Lowest heart rate during the period

4.5.3 Distance Summary

Distance summary provides the distance and time related metrics (speed, pace, et al) for an activity. This data is calculated from sensors.

Property Name	***Type***	***Description***
Period	Object	Length of time bucket; see Period
TotalDistance	Long	Total distance in the period
TotalDistanceOnFoot	Long	Total distance covered on foot
ActualDistance	Long	Absolute distance, including any paused time
ElevationGain	Int	Cumulative elevation gain during the period
ElevationLoss	Int	Cumulative elevation loss during the period
MaxElevation	Int	Maximum elevation during the period
MinElevation	Int	Minimum elevation during the period
WayPointDistance	Long	Distance in cm used to waypoint the GPS data
Speed	Int	Total Period distance divided by period duration
Pace	Int	Period duration divided by total period distance
OverallPace	Int	Duration of all periods divided by distance of all periods

4.5.4 Performance Summary

Performance summary captures the heart rate at the end of the activity as well data about heart rate zones.

4.5.4.1 JSON Representation

```
"performanceSummary": {
  "finishHeartRate": 152,
  "recoveryHeartRateAt1Minute": 123,
  "recoveryHeartRateAt2Minutes": 110,
  "heartRateZones": {
    "underAerobic": 1,
    "aerobic": 5,
    "anaerobic": 4,
    "fitnessZone": 6,
    "healthyHeart": 1,
    "redline": 1
  }
```

4.5.4.2 Properties

Property Name	***Type***	***Description***
FinishHeartRate	Object	Heart rate when user finished the activity
HeartRateZones	Object	Summary of heart rate zones, see HeartRate Zones
RecoveryHeartRateAt1Minute	Duration	Heart rate at one minute after the activity is finished
RecoveryHeartRateAt2Minutes	Long	Heart rate at two minutes after the activity is finished

4.6 PERIOD

Period defines the duration in which to display the summary. The currently supported values are:

Hourly	Divides the day(s) into hourly buckets and shows the data per hour
Daily	Summarizes the data on daily basis
Minute	Summarizes the data on a minute basis

Activity	Summarizes the data on activity basis

4.7 API REQUEST PARAMETERS

4.7.1 Profile

Resource URL: <version>/me/Profile

4.7.2 Devices

Resource URL

- <version>/me/Devices
- <version>/me/Devices/{Deviceid}

4.7.3 Summaries

Resource URL

- <version>/me/Summaries/{period}?startTime={StartTime}&endTime={endTime}&deviceIds={deviceIds}&maxPagesize={maxPagesize}

Date ranges are start time plus two days for hourly summaries. There is no range restriction for daily summaries.

Parameters

Arguments	**Details**
Period (required)	Daily or Hourly
startTime (required)	Start time for data. Start time is inclusive.
endTime (optional)	End time for data. End time is exclusive.
deviceIds (optional)	Filter for particular device id
maxPageSize (optional)	Maximum page size

4.7.4 Activities

Resource URL

- <version>/me/Activities/{ActivityIds}?startTime={startTime}&endTime={endTime}&activityIncludes={activity includes}&activityTypes={activitytypes}&deviceIds={deviceIds}&splitDistanceType={splitDistanceType}&maxPageSize={maxPageSize}
-

Parameters

Arguments	Details
ActivityIds	List of particular ids to get
startTime (required)	Start time for data. Start time is inclusive.
endTime (required)	End time for data. End time is exclusive.
activityTypes (optional)	Filtering for activity types using comma-separated list. Values are • Run • Sleep • FreePlay • GuidedWorkout • Bike • Golf
activityIncludes (optional)	Provides additional details for each activity. Values are using comma-separated list: • Details: provides breakdown of activity into segments • MinuteSummaries: provides data at each minute interval; see Minute Details for details • MapPoints: provides breakdown of activity per distance intervals; see Map Points for details
deviceIds (optional)	Filter for particular device id using comma-separated list
splitDistanceType (optional)	Distance; options are miles or kilometers. If not specified, miles is default.
maxPageSize (optional)	Maximum page size

Special Notes

When filtering on particular activity ids, please note the following:

- If other parameters are applied that exclude the activity (i.e. a date range is passed in which is outside of where the activity id is) the results will come back empty
- Results may be unpredictable if there is a mismatch between activity type and activityid (i.e. you pass in an activity id of a sleep, but pass in activityType=Run)

4.8 Application Authentication Scheme and Workflow

The work flow for getting the access token is as follows.

Step 1. Get an authentication code

To start the sign-in process within your application or web service, use a web browser or web browser control to load a URL request.

Request

```
GET https://login.live.com/oauth20_authorize.srf?client_id={client_id}
&scope={scope}&response_type=code&redirect_uri={redirect_uri}
```

Parameter name	Value	Description
client_id	string	The client ID created for your app.
scope	string	A space-separated list of authorization scopes that your app requires.
redirect_uri	string	The redirect URL that the browser is sent to when authentication is complete. Use this redirect URL for mobile and desktop applications: https://login.live.com/oauth20_desktop.srf

Response

Upon successful authentication and authorization of your application, the web browser will be redirected to the redirect URL with additional parameters added to the URL.

```
https://login.live.com/oauth20_authorize.srf?code=df6aa589-1080-b241-b
410-c4dff65dbf7c
```

Step 2. Redeem the code for access tokens

After you have received the code value, you can redeem this code for a set of tokens that allow you to authenticate with the Microsoft Health Cloud API.

Request

To redeem the code, make the following request:

```
POST https://login.live.com/oauth20_token.srf?client_id={client_id}&re
direct_uri={redirect_uri}&client_secret={client_secret}&code={code}&gr
ant_type=authorization_code
```

The request body is a properly encoded URL string, with some required parameters.

Parameter name	Value	Description
client_id	string	The client ID value created for your application.

Parameter name	Value	Description
redirect_uri	string	The redirect URL that the browser is sent to when authentication is complete. This should match the redirect_uri in the first request.
client_secret	string	The client secret created for your application.
code	string	The authorization code you received in the first authentication request.

Note: For web apps, the domain portion of the redirect URI must match the domain portion of the redirect URL that you specified in the app.

Response

If the call is successful, the response for the POST request contains a JSON string that includes several properties, including access_token, authentication_token, and refresh_token (if you requested the offline_access scope).

```
{
  "token_type":"bearer",
  "expires_in": 3600,
  "scope":"mshealth.ReadDevices",
  "access_token":"EwCo...AA==",
  "refresh_token":"eyJh...9323"
}
```

Response Parameter name	Value	Description
token_type	string	Authorization type: “Bearer” in this case
expires_in	long	The amount of time in seconds when the access token is valid. You can request a new access token by using the refresh token (if available), or by repeating the authentication request from the beginning.
scope	string	A space-separated list of scopes that your app requires.
access_token	string	Access token to authenticate against Microsoft Health Cloud APIs

Response Parameter name	Value	Description
refresh_token	string	The refresh token you received previously

Step 3. Calling Microsoft Health Cloud API

You can now store and use the access_token provided to make authenticated requests to the Microsoft Health Cloud API.

```
GET https://api.microsofthealth.net/v1/me/Profile/

Authorization: bearer EwCo...AA==
```

Step 4. Get a new access token or refresh token (optional)

If your app has requested access to offline_access, this step will return a refresh_token that can be used to generate additional access tokens after the initial token has expired.

Request

To redeem the refresh token for a new access token, make the following request:

```
GET https://login.live.com/oauth20_token.srf?client_id={client_id}&redirect_uri={redirect_uri}&client_secret={client_secret}&refresh_token={refresh_token}&grant_type=refresh_token
```

The request body is a properly encoded URL string, with some required parameters.

Parameter name	Value	Description
client_id	string	The client ID created for your application.
redirect_uri	string	The redirect URL that the browser is sent to when authentication is complete. This should match the redirect_uri value used in the first request.
client_secret	string	The client secret created for your application.
refresh_token	string	The refresh token you received previously.

Note: For web apps, the domain portion of the redirect URL must match the domain portion of the redirect URL that you specified in the application.

Response

If the call is successful, the response for the POST request contains a JSON string that includes several properties, including access_token, authentication_token, and refresh_token (if you requested the offline_access scope).

```
{
  "token_type":"bearer",
  "expires_in": 3600,
  "scope":"mshealth.ReadDevices",
  "access_token":"EwCo...AA==",
  "refresh_token":"eyJh...9323"
}
```

Response Parameter name	**Value**	**Description**
token_type	string	Authorization type: “Bearer” in this case
expires_in	long	The amount of time in seconds when the access token is valid. You can request a new access token by using the refresh token (if available), or by repeating the authentication request from the beginning.
scope	string	A space-separated list of authorization scopes that your app requires.
access_token	string	Access token to authenticate against Microsoft Health Cloud APIs
refresh_token	string	The refresh token you received previously

Step 5. Log Out (Optional)

To sign a user out, perform the following steps:

1. Delete any cached access_token or refresh_token values you've previously received from the OAuth flow.
2. Perform any sign out actions in your application (for example, cleaning up local state, removing any cached items, etc.).
3. Make a call to the authorization web service using this URL:

```
GET https://login.live.com/oauth20_logout.srf?client_id={client_id}&redirect_uri={redirect_uri}
```

This call will remove any cookies that enable single sign-on to occur and ensure that next time your app launches the sign in experience, the user will be requested to enter a username and password to continue.

Parameter name	Value	Description
client_id	string	The client ID created for your application.
redirect_uri	string	The redirect URL that the browser is sent to when authentication is complete. This should match the redirect_uri value used in the first request.

After removing the cookie, the browser will be redirected to the redirect URL you provided. When the browser loads your redirect page, no authentication query string parameters will be set, and you can infer the user has been logged out.

5 SAMPLE APPLICATION

There is a sample Windows application available to help you get starting with the Microsoft Health Cloud API. To use the sample Windows application, you will need:

- Visual Studio 2013 Ultimate with Update 4, which is available at http://www.microsoft.com/en-US/download/details.aspx?id=44915
    - During installation, please select "Windows SDK 8.1" to install the Windows SDK.

To download and run the sample Windows application:

1) Navigate to: http://developer.microsofthealth.com/CloudAPI
2) Click on the “Sample Application” link to download MicrosoftHealthSample.zip. Extract all files in the zipped folder:
    - MicrosoftHealthSample
        - MicrosoftHealthSample.Shared
        - MicrosoftHealthSample.Windows
        - MicrosoftHealthSample.WindowsPhone
    - MicrosoftHealthSample.sln
3) Open Visual Studio 2013.
4) Select Open Project, browse to your local folder and select MicrosoftHealthSample.sln
5) You will be prompted to sign in and get a developer license the first time if you don't already have one.
6) Drop down the View menu and select Solution Explorer.
7) In the Solution Explorer panel, open MicrosoftHealthSample.Shared/MainPage.cs at the bottom of the list and update the clientId and clientSecret strings.
   The clientId and clientSecret can be obtained from

https://account.live.com/developers/applications. Refer to the Account Creation and Application Registration for more information.

8) Build and run the solution.
9) Click "Sign In" and enter the username and password for the Microsoft Account associated with your Microsoft Health data.
10) Use the buttons to view your Microsoft Health data.
11) Please note that "Get Profile" is intentionally broken to demonstrate the error returned if the user has not authorized the right scope.

6 SUPPORT

Please reach out to mshealth@microsoft.com for issues.