Capabilities of Garmin devices in monitoring exercise effectiveness

Sensors Available:

* Heart Rate Monitor
* Barometric Altimeter
* Compass
* Gyroscope
* Accelerometer
* Thermometer
* Pulse OX Blood Oxygen Saturation Monitor
* GPS
* GLONASS – Russian satellite navigation system, providing real time position and velocity determination for military and civilian users
* GALILEO – Europeans global satellite navigation system, provides accurate GPS service under civilian control

App:

* Garmin Connect – allows visualization of collected stats from Garmin devices

Frameworks:

* Flexible and Interoperable Data Transfer SDK (FIT)

FIT is designed specifically for the storing and sharing of data that originates from sport, fitness, and health devices. The protocol defines a set of data storage templates (FIT messages) that can be used to store information such as user profiles, activity data, courses, and workouts. It is specifically designed to compact, interoperable and extendible.

* Garmin Health SDKs

Available to Garmin Health enterprise partners, the SDKs allow for direct integration of Garmin wearables in your Android or iOS app.

* Standard SDK:

Allows access to all health and fitness activity data directly from your mobile app without the need for web service integration.

* Companion SDK:

Allows access to real-time sensor streams from Garmin wearables. Get instant access to current activity data, like step counts, or subscribe to live streams of heart rate, stress scores accelerometer, and more

Types of sensor data classes available from API:

Sensor.AccelerometerData (API 2.3.0):

The values for the x, y and z axis are in Milli G units. For reference 1000 Milli G = 1 G. This appears to combine Gyroscope as well within the class used to store accelerometer data.

* Pitch: Array<Float> -> The Array of pitch values as Floats in degrees
* Power: Array<Number> -> The Array of vector power values as Numbers in millig-units
* Roll: Array<Float> -> The Array of roll values as Floats in degrees
* X: Array<Number> -> The Array of x axis values as Numbers in millig-units
* Y: Array<Number> -> The Array of y axis values as Numbers in millig-units
* Z: Array<Number> -> The Array of z axis values as Numbers in millig-units

Sensor.HeartRateData (API 3.0.0):

Provides granular hear rate data like beat-to-beat intervals.

* heartBeatIntervals: Array<Number> -> the most recent beat-to-beat interval data as an Array of Number objects in milliseconds (ms).

Sensor.Info (API 1.0.0 – 3.2.0):

Contains all the information provided by enabled sensors.

* Accel: Array<Number> -> The accelerometer reading of the x, y, and z axes as an Array of Number values in millig-units
* Altitude: Float -> The altitude above mean sea level in meters (m)
* Cadence: Number -> The cadence in revolutions per minute (rpm)
* Heading: Float -> The true north referenced heading in radians
* heartrate: Number -> The heart rate in beats per minute (bpm)
* mag: Array<Number> -> The magnetometer reading of the x, y, and z axes as and Array of Number values in milliGauss (mG)
* oxygenSaturation: Number -> The current oxygen saturation in percent (%) (API 3.2.0)
* power: Number -> The power in Watts (W)
* pressure: Float -> The barometric pressure in Pascals (Pa)
* temperature: Float -> The temperature in degrees Celsius (C)

Position.Info (API 1.0.0)

Contains all the information provided by the positioning system

* accuracy: Position.Quality -> the positional accuracy
* altitude: Float -> the elevation above mean sea level in meters (m)
* heading: Float -> The true north referenced heading in radians
* position: Position.Location -> The latitude and longitiude of the location
* speed: Float -> The horizontal speed in meters per second (mps)
* when: Time.Moment -> The GPS time stamp of the obtained Location fix

Position.Location (API 1.0.0):

The location object represents a specific position. Location objects provide methods for retrieving position coordinates in various formats

* latitude: Number -> The latitude
* Longitude: Number -> The longitude
* Format: Land.Symbol -> The format of the provided latitude and longitude as one of three possible values:
  + :degrees
  + :radians
  + :semicircles

ActivityMonitor.ActiveMinutes (API 2.1.0):

A class representing the active minutes recorded by the device

* Moderate: Number -> The total number of moderate activity minutes recorded by the device
* Total: Number -> The total number of active minutes recorded by the device
* Vigorous: Number -> The total number of vigorous activity minutes recorded by the device

ActivityMonitor.HeartRateSample (API 1.2.2)

A class containing heart rate data at a given time

* heartrate: Number -> Heart rate in beats per minute (bpm)
* when: Time.Moment -> The time of the heart rate sample

ActivityMonitor.History (API 1.0.0 – 2.1.0)

A class containing information about the user’s activity history

* activeMinutes: ActivityMonitor.ActiveMinutes -> Number of active minutes
* calories: Number -> The calories for the day in kilocalories (kCal)
* distance: Number -> The distance for the day in centimeters (cm)
* floorsClimbed: Number -> The number of floors climbed for the day
* floorsClimbedGoal: Number -> Floors climbed goal for the day
* floorsDescended: Number -> The number of floors descended for the day
* startOfDay: Time.Moment -> A moment object representing the start time of the day recorded by this History object
* stepGoal: Number -> The step goal for the day in number of steps
* steps: Number -> The step count for the day in number of steps

ActivityMonitor.Info (API 1.0.0)

Contains information about the user’s current activity status. NOTE: unless otherwise indicated, Info data is calculated for the current day starting at midnight as provided by the device

* activeMinutesDay: ActivityMonitor.ActiveMinutes -> The number of active minutes for the current day
* activeMinutesWeek: ActivityMinitory.ActiveMinutes -> The number of active minutes for the current week
* activeMinutesWeekGoal: Number -> The value of the active minutes goal for the current week
* calories: Number -> The calories burned so far for the current day in kilocalories (kCal)
* distance: Number -> The distance since midnight for the current day in centimeters (cm)
* floorsClimbed: Number -> the number of floors climbed for the current day
* floorsClimbedGoal: Number -> the current floor climb goal
* floorsDescended: Number -> The number of floors descended for the current day
* metersClimbed: Float -> The vertical distance of floors climbed in meters (m)
* metersDescended: Float -> The vertical distance of floors descended in meters (m)
* respirationRate: Number -> Current respiration rate for the user, in breaths per minute
* stepGoal: Number -> The step goal for the current day in number of steps
* steps: Number -> The step count since midnight for the current day in number of steps
* timeToRecovery: Number -> Time to recovery from the last activity, in hours

Activity.Info (API 1.0.0)

Contains all the information about the current activity. There are too many instance members to list here but they can be viewed at:

<https://developer.garmin.com/connect-iq/api-docs/Toybox/Activity/Info.html>

UserProfile.Profile (API 1.0.0 – 3.3.0)

The Profile object contains user information (List below is a subset of full instance members)

* averageRestingHeartRate: Number -> The user’s seven-day average resting heart rate (bpm)
* restingHeartRate: Number -> Heart rate in beats per minute (bpm)
* runningStepLength: Number -> Running step length in millimeters (mm)
* vo2maxCycling: Number -> The user’s VO2 Max value for cycling activity
* vo2maxRunning: Number -> The user’s VO2 Max value for running activity
* walkingSetpLength: Number -> Walking step length in millimeters

UserProfile.UserActivity (API 1.0.0)

A class for storing user activity information

* distance: Number -> Distance covered by activity in meters
* duration: Number -> Duration of the activity in seconds
* startTime: Time.Moment -> Start time of the activity
* type: Number -> Sport type of the activity

Weather.CurrentConditions (API 3.2.0)

Represents the most recently cached weather conditions

* condition: Number -> The current weather condition
* feelsLikeTemperature: Number -> The wind chill or heat index, in Celsius
* highTemperature: Number -> The forecasted high temperature for the day in Celsius
* lowTemperature: Number -> The forecasted low temperature for the day in Celsius
* observationLocationName: String -> Textual description of the observation location
* observationLocationPosition: Position.Location -> Location where the conditions were observed
* observationTime: Time.Moment -> UTC time the conditions were observed
* precipitationChance: Number -> The chance of precipitation [0-100%]
* relativeHumidity: Number -> The relative humidity [0-100%]
* temperature: Number -> The current temperature in Celsius
* windBearing: Number -> The wind bearing in degrees
* windspeed: Float -> The current wind speed in meters per second

Cycling and compatibility with devices like Shimano Di2:

* Shimano is one of the leading bike parts manufacturers and many higher end bikes typically integrate a lot of Shimano parts.
* Shimano Di2 can found on some of the Dura Ace and Ultregra group-sets and works via Bluetooth to allow the rider to change gears without the need of cables connecting the handlebar and derailers.
* The rider would change a gear and the shifter would send a message to the front or rear derailer and push the chain onto the correct gear
* Di2 products are compatible with certain Garmin devices as well as smartphones and can deliver relevant ride information to the user in real time
* I imagine that some of the cycling information from the API would come from devices like these

Exporting sensor data from the device:

Graphical user interface, text, application, email

Description automatically generated

Graphical user interface, text, application, email

Description automatically generated

Graphical user interface, text, application, email

Description automatically generated

Suitable devices to use:

* Fenix series
* Epix (Gen 2)
* Forerunner series
* Venu Series
* Vivoactive 4

Sources:

* <https://www.esa.int/Applications/Navigation/Galileo/What_is_Galileo>
* <https://developer.garmin.com/connect-iq/api-docs/index.html>
* <https://www.glonass-iac.ru/en/>
* <https://www.smartwatchspecifications.com/devices/garmin-fenix-7-smartwatch-specs-review/>
* <https://bike.shimano.com/en-AU/technologies/component/details/di2.html>
* <https://developer.garmin.com/fit/overview/>
* <https://developer.garmin.com/health-sdk/overview/>
* <https://support.garmin.com/en-US/?faq=W1TvTPW8JZ6LfJSfK512Q8>